

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: N2612 Elektrotechnika a Informatika

Studijní obor: 1892T007 Informační technologie

Řídicí jednotka elektromobilu

Control unit for electric vehicle

Diplomová práce

Autor: **Antonín Puncman**

Vedoucí práce: Ing. Pavel Jandura

V Turnově 1.5. 2013

Na tomto místě bude oficiální zadání práce

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje Zákon č. 121/2000 Sb. o právu autorském, zejména § 60 - školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne úhradu nákladů, které vynaložila na vytvoření díla, až do její skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce.

Datum:

Podpis:

Abstrakt

Cílem této diplomové práce je pochopení a přiblížení elektromechanických komponent elektromobilu, analýza možností snímání, zisku a zobrazení jízdních a palubních údajů, a to i s ohledem na dostupné komponenty dané konstrukcí. Následně se projekt zabývá definicí požadavků na řídicí jednotku, a z ní vyplývajících specifikací sběrnici (CAN, RS232), a periférií jednotkou ovládaných. V další části projekt rozhoduje volbu mikrokontroléru AT90CAN128, vhodného pro komunikaci s danými perifériemi. V kapitolách o hardwaru je dokumentována volba zobrazovacích prvků, volba nových a úprava konstrukcí daných ovládacích prvků. Navazující dokumentovaná část tohoto projektu zaznamenává vývoj, navržení, sestavení a zapojení celé jednotky. Programování a testování pak zahrnuje kompletní fyzickou instalaci a přípravu na vozidle, včetně úpravy stávajících komponent. V závěru práce pojednává o dalších možnostech rozšíření jednotky a definuje nové podklady a požadavky pro další generaci řídicích jednotek pro elektromobil.

Klíčová slova

- AT90CAN128
- CAN
- RS232
- WHEEL SENZOR
- ELEKTROMOBIL

Abstract

The aim of this master's thesis is to understand and zoom in electric and electromechanical components of electric vehicle, analysis of possibilities of scanning gaining and displaying mounted and on-board data, and also with regard to the available components determined by construction. Subsequently, the project addresses the definition of requirements for the control unit and the ensuing specification bus (CAN, RS232) and peripherals controlled by unit. In another part of the project determines the choice of microcontroller AT90CAN128, suitable for communication with the determined peripherals. In chapters about hardware is documented choice of displaying elements, the choice of new and modifications of set controls. Follow-documented part of this project records the development, design, construct and involvement of the entire unit. Programming and testing involves a complete physical installation and preparation of the vehicle, including modifications components. The conclusion discusses other options for expansion units and defines new ideas and requirements for the next generation of control units for electric vehicle.

Key words

- AT90CAN128
- CAN
- RS232
- WHEEL SENZOR
- ELECTRIC VEHICLE

Obsah

| | |
|---|----|
| Titulní strana..... | 1 |
| Originální zadání práce..... | 2 |
| Prohlášení..... | 3 |
| Abstrakt | 4 |
| 1. Úvod | 8 |
| 2. Elektromobil..... | 10 |
| 2.1 Pohon..... | 11 |
| 2.2 Baterie, nabíječ | 12 |
| 2.3 Čidla..... | 13 |
| 2.4 Spínací skříňka..... | 14 |
| 2.5 Souhrn požadavků..... | 14 |
| 3. Rychloměr 1.0..... | 17 |
| 3.1 Čidla..... | 18 |
| 3.2 Schéma rychloměru..... | 19 |
| 3.3 Výpočet rychlosti a naprogramování..... | 20 |
| 4. Gen 2.0 - hardware..... | 22 |
| 4.1 Blokové schéma..... | 23 |
| 4.2 MCU..... | 24 |
| 4.3 Obsluha nabíječe..... | 30 |
| 4.4 RS-232..... | 32 |
| 4.5 Start..... | 33 |
| 4.6 CAN..... | 35 |
| 4.7 Senzory..... | 36 |
| 4.8 Zobrazovací modul..... | 37 |
| 4.9 Realizace..... | 41 |
| 4.9.1 Prototyp..... | 41 |
| 4.9.1 Spínací skříňka..... | 49 |

| | | |
|-----|---|----|
| 5. | Gen 2.0 - software | 51 |
| 5.1 | Programovací prostředí | 51 |
| 5.2 | Nastavení mikrokontroléru | 51 |
| 5.3 | Zdrojové kódy | 51 |
| 5.3 | CAN - regulátor | 51 |
| 6. | Závěr | 53 |
| | Seznam použité literatury | 55 |
| | Seznam obrázků | 58 |
| | Příloha A – Schéma zapojení rychloměru 1.0 | 60 |
| | Příloha B – Schéma zapojení zobrazovacího modulu | 61 |
| | Příloha C – Schéma zapojení centrálního modulu jednotky | 62 |
| | Příloha D – Hlavní část zdrojového kódu | 63 |
| | Příloha CD s elektronickou podobou práce | CD |

1. Úvod

Jak již napovídá název práce, na následujících stranách bude popsána jedna z možností, jak realizovat část řídicí elektroniky, ve světě jednoho z mnoha, ale prvního elektromobilu postaveného na Technické univerzitě v Liberci. Onen hlavní titul celého projektu, nad-téma chcete-li, elektromobil, je dle mého nejhlubšího přesvědčení nejen tématem budoucnosti, bohužel globální síly, zaměřené na maximalizaci finančního zisku, jej v řádu let intenzivně zašlapávají již od projektu GM EV1 [4] z 90tých let minulého století průběžně, do minulosti.

K motivaci účasti na tomto projektu. Již při volbě vysoké školy na gymnáziu jsem jako zapálený nadšenec do technického pokroku všeho druhu především pak automobilistické odnože, uvažoval své školní, mimoškolní a při troše štěstí následné absolventské působení v oblasti vývoje automobilů.

V průběhu studia pak toto působení probíhalo především v garážovém prostředí na automechanické úrovni prostřednictvím oprav, renovací modernizací a vlastně i kompletní stavby automobilu. Studium na Fakultě mechatroniky pak přineslo znalosti a zkušenosti potřebné pro úpravy řídicí elektroniky motoru, například pro adaptaci na alternativní palivo E85, či zvýšení efektivity spalování sekvenčním vstřikováním paliva místo vstřiku silmultálního.

Účast na takovém projektu, jako je sestrojení provozuschopného elektromobilu, je pro mě jedinečnou příležitostí k předvedení a uplatnění dovedností získaných v minulých letech, ať už jistou mutací reverzního inženýrství v garáži, či tvorbou školních projektů a prací. S nadšením a přáním trochu přispět k technickému pokroku a popularizaci elektromobility v soupeření s ekonomicko-maximálně-ziskovými argumenty jsem přijal toto, především z hlediska komplexnosti a spolupráce s další fakultou, nesnadné téma elektronické jednotky za své.

Úkoly této magisterské práce se dají rozčlenit na tři hlavní podskupiny. Identifikace vstupů a výstupů a jejich následná analýza, návrh jednotky a nakonec fyzická realizace a testování.

Analytická část začíná již prvotní problematikou vývoje, udávající některé podklady pro návrh jednotky, a to synchronizací vývoje konstrukce a elektroniky. Před konstrukcí vozidla do nekonkrétních požadavků na řídicí jednotku připadá volba snímání některých fyzikálních veličin pohybu vozidla, sledování otáčení kol, palubních údajů

a možnost komunikace s elektronikou pohonné jednotky. Předpokládána též byla schopnost zobrazování vybraných údajů a jejich ukládání.

Následné zkonstruování vozu použitím konkrétních podvozkových komponent, ovládacích prvků a tvarů v kombinaci s konkrétní elektronikou pohonu požadavky na řídicí jednotku redefinuje a konkretizuje do jasnější formy.

Konstrukce vozu používající podvozkové díly vozu Škoda Fabia/Roomster vyžaduje schopnost řídicí jednotky vyhodnocovat koncernová čidla otáčení kol a zpracovávat cenné údaje o otáčení kol. Tvaru a ergonomii vozidla zároveň podléhají návrhové i realizační práce, neb pro prototyp jednotky je za volantem jen minimální prostor.

Návrh řídicí jednotky pak řeší podmínku komunikace s BMS (battery management system), na jejímž základě ovládá nabíjení hlavních pohonných baterií. Zajímavá data a údaje o provozu se získávají komunikací s pokročilým efektivním regulátorem pohonného elektromotoru. Zásadní role jednotky a důležitý úkol práce je pak napojení na tovární spínací skříňku z vozu Škoda Fabia tak, aby přepínala nabíjecí/pohotovostní, antispark a provozní režim a napodobovala provoz běžného automobilu. Pro experimentální část celého elektromobilního projektu pak tento podprojekt zahrnuje řešení měření volitelných údajů AD převodníky a možnost ukládání volitelných dat na přenosnou paměť.

Sledem zásad pro vypracování, podrobnější popisy, analýza, návrhy řešení a řešení samotné následuje v dalších kapitolách.

2. Elektromobil

Pro lepší představu a ucelení problematik je vhodné si elektromobil letmo popsat. Konstrukce dvoumístného draku vozu, v uspořádání vedle sebe, jak je patrné na obrázku 2.1, byla zvolena experimentální, sestavovaná převážně ze šroubovaných hliníkových profilů, užívaných do té doby k modulární stavbě výrobních linek a stojanů. Tento drak je osazen na míru vyrobeným ocelovým uložením dvou předních náprav z vozu Škoda Fabia/Roomster, vpředu řízená s atypickými zkrácenými tlumiči a pružinami, v zadu aretovaná

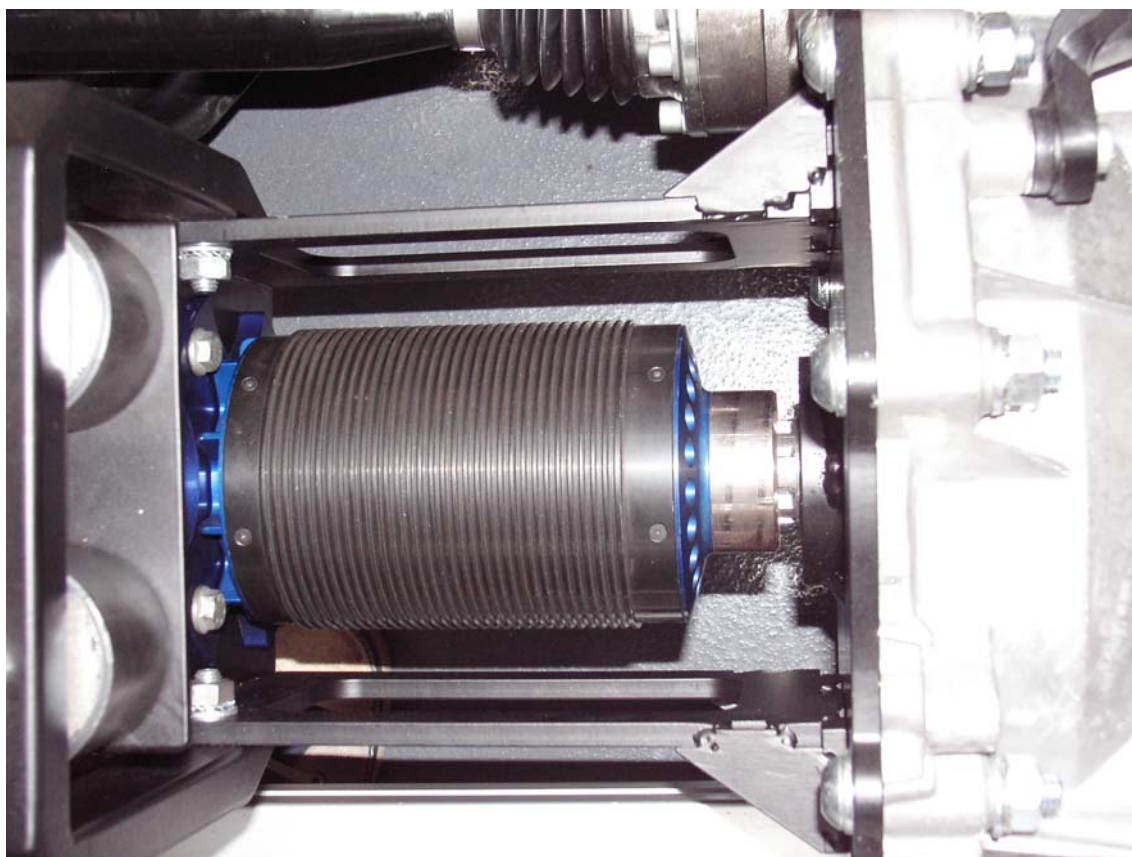
Pohonná jednotka sestávající se z experimentálního modelářského elektromotoru a koncernové převodovky MQ200 je umístěna vpředu a pohání přední nápravu. O přísun energie k motoru se stará programovatelný střídavý regulátor a sada bateriových článků, rozdělených na dva oddíly, které jsou rozmístěny do přední a zadní části vozu pro lepší rozložení hmotnosti vozidla. Řízení hřebenové bez posilovače, brzdy kotoučové chlazené 288mm vpředu, 256mm vzadu, s dvoukruhovým zapojením přední/zadní náprava taktéž bez posilovače. Nyní detailnější popis komponent.



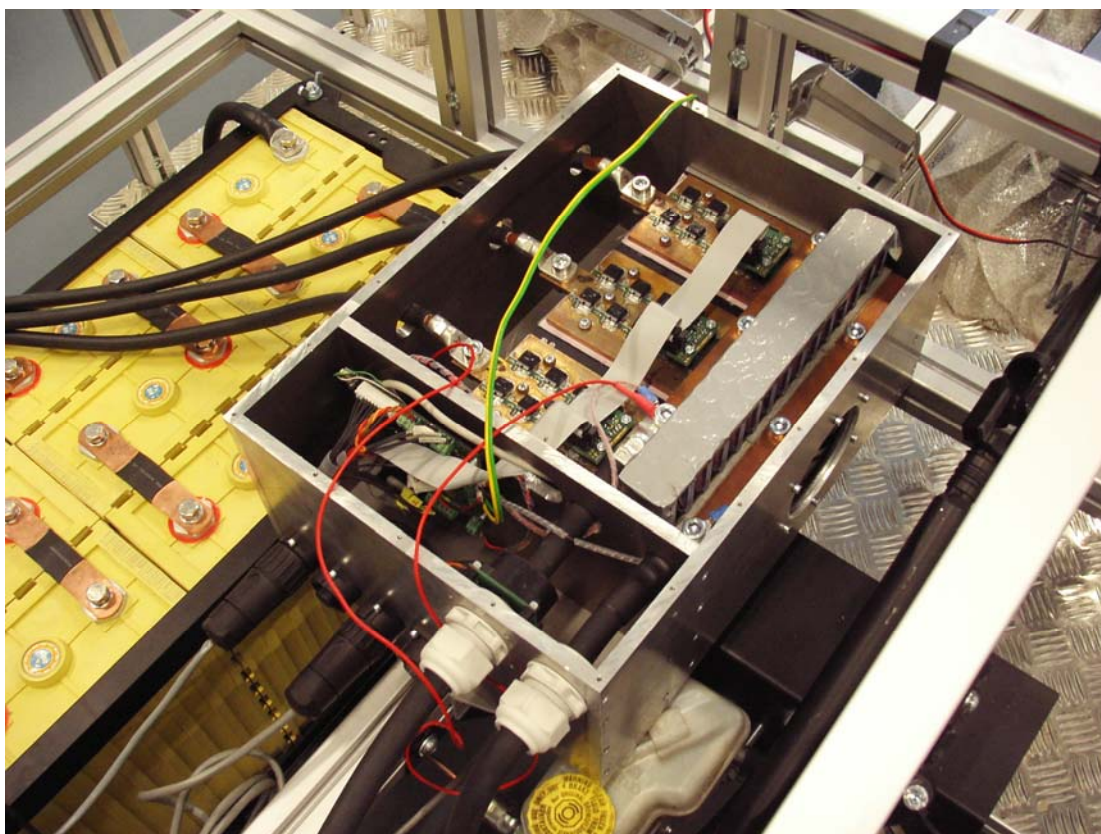
Obr. 2.1 : Stavba elektromobilu e-Šus

2.1 Pohon

Začneme pohonným ústrojím. Hlavním pokrokovým řešením je užití BLDC (Brushless DC) technologie. Ta je převzata z experimentálního elektromobilu Solarex konstruktérů Borise Guiče a Luboše Chvátala. Jde o pohon tvořený elektromotorem s výkonem až 40kW a jeho regulací. Motor je synchronní bezkartáčkový el. motor [5] s elektronickou komutací, neodymovými magnety a rotačním pláštěm (obr. 2.2). Na konstrukci tohoto typu motoru pracují firmy Carco a FreeAir již od roku 2000. Řešení regulace [6] (obr. 2.3) motoru dodává česká firma MGM Compro ze Zlína, která se regulací průmyslových aplikací zabývá a která je schopna a ochotna regulátory vyšších výkonů pro větší BLDC motory vyvinout a vyrobit. Výhody tohoto druhu pohonu jsou vysoká elektrická účinnost při malých rozměrech a hmotě jak motoru tak regulátoru. Dojezd Solarexu inspirující náš elektromobil je dle použitých baterií (90 kg u prezentované verze) asi 100 km při průměrné rychlosti 50 km/hod.



Obr. 2.2 : BLDC motor HCS C 40000W Industry



Obr. 2.3 : Regulátor MGM Compro HBC-40063

K sepnutí okruhu napájení regulátoru je určeno výkonné relé Kilovac EV200 firmy Tyco Electronic [7], spodobněné na obr. 2.4. Jeho základní parametry pro přenášené napětí až 900V a pro proud až 500A bezpečně splňují naše požadavky, výhodou je též verze s ekonomizérem.



Obr. 2.4 : EV200

2.2 Baterie, nabíječ

O čem to celé je aneb baterie. Jak jsem se dočetl, i k vývoji pohonu konstruktéry Solarexu inspirovala především dostupnost nových bezpečných lithiových akumulátorů na bázi LiFePO₄. V našem případě se jedná o sadu sedmnácti článků LiFeYPO₄ (LFP s příměsí yttria) dovozce GWL/Power typ WB-LYP200AHA[9] LiFeYPO₄ (3.2V/200Ah) (obr.2.5). Dle informací dodavatele je



nominální napětí LiFePO₄ článků je zhruba 3,2 V **Obr. 2.5 : 3.2V/200Ah [8]** (mezi 3,0 až 3,3 V), maximální nabíjecí napětí při prvním startovním nabití je 4V.

Doporučené následné nabíjení je do 3,8V. Minimální napětí, na které je možné články vybit, je 2,5V. Tyto články jsou schopny dodat pracovní proud impulsně až 20C (tj. 20x kapacita), dlouhodobě až 3C. Životnost je udávána až 2000 cyklů. Lze je použít v rozmezí teplot -30°C až +65°C. Samovybíjení u těchto článků prakticky neexistuje, a pro zanedbatelný paměťový efekt je možné je nabíjet prakticky v jakékoli části cyklu.

Palubním příslušenstvím k bateriím je nabíječ typu POW48V30A/BMS [10] (obr. 2.6). Jedná se o nabíječ z nabídky dodavatele našich LiFeYPO4



článků, vstupní napájením je AC 230V. Nabíječ disponuje přepětovou ochranou, časovou ochranou, při

POW 48V 30A
+ BMS option



Obr. 2.6 : POW48V30A [8]

selhání nabíjení ochranou proti zkratu a přepólování. Průběh nabíjení je indikován prostřednictvím LED. Náš model je osazen konektorem pro připojení Battery-

Obr. 2.7: relé RL709-48V [8] management systému. S funkcí nabíječe je též spřáhnuta možnost odpojení baterie od nabíječe prostřednictvím bistabilního relé RL709-48V [11] (obr. 2.7).

2.3 Čidla

Další elektromechanickou komponentou dodanou konstrukcí nutnou uvažovat, jsou díky použití koncernových kompletů náprav čidla otáčení kol od výrobce Ate (obr. 2.8). Čidla jsou dle měření



Obr. 2.9 : Magnetický kroužek

totožná, leč rozdílná

tvarově dle umístění **Obr. 2.8 : ABS sensor ATE** v pravé či levé těhlici kola. Levé s referenčním číslem Ate 24.0711-5273.3 a pravé 24.0711-5274.3. O principu fungování těchto čidel se toho dá z oficiálních zdrojů dozvědět jen málo.



Z neoficiálních zdrojů, odhadů a měření se pak jedná o dvojici unipolárních Hallových senzorů s logikou měnící úbytek napětí na výstupu

senzor. Párovou částí těchto snímačů je pak na každém ložisku kola magnetický kroužek (obr. 2.9) se čtyřiceti dvěma póly, který otáčením v těsné blízkosti čidla překlápí jeho logickou úroveň.

2.4 Spínací skříňka

Poslední důležitou elektromechanickou komponentou je dodaná koncernová spínací skříňka z vozu Škoda Fabia (obr. 2.10). Skříňka je dimenzovaná na spínání 12V napětí a proudů v řádu jednotek až desítek A. Disponuje třemi polohami. Třetí poloha spíná u spalovacího motoru startér a je vracena pružinou do polohy druhé, ve které je bez vrácení do polohy první



Obr. 2.10 : Spínací skříňka

blokována vodící kulisou. První poloha pak má odlišné sepnutí jednoho z kontaktů při otočením klíče po zasunutí oproti navrácení klíče z polohy druhé. Detailnější rozbor a následné úpravy v kapitolách návrhů a realizace.

2.5 Souhrn požadavků

Po seznámení se s elektrickými a elektromechanickými komponenty elektromobilu je v první části této podkapitoly na řadě definice požadavků na řídicí jednotku plynoucí z parametrů jednotlivých komponent a jejich vlastností, které bych rád přiblížil.

Podkapitola 2.1 o pohonu nám dává k dispozici programovatelný střídavý regulátor, na nějž se zaměříme. Disponuje totiž sběrnici CAN, jejíž rozhraní je pro řídicí jednotku vyžadované z následujícího důvodu. Na Sběrnici jsou vysílané informační pakety periodicky bez nutnosti vyžádání či potvrzení přijímací stranou. Komunikace s regulátorem nám tak dává k dispozici hned několik údajů o palubní elektronice a jízdě jako napětí z baterie, proud z baterie, otáčky motoru, odometr (počet otáček od zapnutí), teplota regulátoru, motoru a baterie, vstupní a výstupní PWM a výčet chybových varovných a chybových stavů. Detailní popis komunikace v dalších kapitolách.

Hlava 2.2 přibližující baterie a nabíječ udává požadavek na jednotku v podobě 4 portů obsluhující režimy nabíjení baterie. Nabíječ lze kombinací uzemnění jednotlivých pinů uvést do stavu vypnuto/zapnuto a maximální nebo redukované

nabíjení. Údaje, na základě kterých je takto činěno, budou získávány z externího BMS zařízení dodávajícím informace prostřednictvím sběrnice RS232, která se tímto zařazuje mezi požadavky.

Základní informací, která nesmí chybět mezi zobrazovanými jízdními údaji, je rychlost vozidla. U předchozích generací vozidel se tak dělo prostřednictvím odečtu otáček výstupu z převodovky, případně otáčení kola stálého převodu diferenciálu přímo v převodovce. Převodovka MQ200 už tuto možnost nenabízí, výpočet rychlosti se u novější generace vozidel s touto převodovkou provádí prostřednictvím přepočtu impulsů senzorů ABS, vznikajících otáčením kol. V kombinaci se senzorem úhlu natočení kol pak lze nejen vypočítávat rychlost, ale i poměr otáčení kol vně a uvnitř zatáčky v případě prokluzu / blokace, a o nastalé situaci informovat brzdové a stabilizační systémy. Náš elektromobil je osazen čidlem na každém kole, tovární snímač úhlu natočení kol je pro jednotku je simulován vlastní konstrukcí. Řídicí jednotka tedy musí umět zpracovat výstupy 4 továrních ABS senzorů, vyhodnocovat je jako okamžitou rychlost vozidla a v kombinaci se senzorem natočení úhlu kol mít možnost upozornit na nesrovnalosti v jejich otáčení.

První část této podkapitoly je ukončena požadavkem na jednotku spínací skříňka a aktivace/deaktivace komponent. Řídicí jednotka musí dle sepnutých poloh a kontaktů správně přepínat provoz jednotlivých částí elektromobilu. Ve vypnutém stavu je žádáno, aby žádná z částí nebyla aktivní a vše bylo odpojeno včetně BMS vstupu nabíječe. V pohotovostním režimu (první poloze) je důležité, aby jednotka odpojila komponenty související s pohonným ústrojím, ale zároveň aby aktivovala a řídila nabíjení. V přechodu do druhé polohy musí deaktivovat nabíjení, v ideálním případě pak rozpojit prostřednictvím relé nabíjecí okruh. Druhá poloha též počítá s přípravou na sepnutí pohonných ústrojí a přivede 12V ke spínacím tranzistorům, ideálně pak spustí napájení řídicí elektroniky regulátoru. V třetí poloze pak jednotka provede spínací sekvenci která, bude mít za úkol sepnutí antispark okruhu integrovaného v jednotce, po nabití filtračních kondenzátorů regulátoru sepnutí hlavního relé a následné rozepnutí antispark okruhu.

A nyní požadavky volné, nesouvisející s komponentami. Jednotka musí umět zobrazovat a prezentovat vybraná data. Vizualních možností je nepřeberně, bude volena jejich kombinace. Pro komplikovanější a volitelná data bude zobrazovací modul používat alfanumerický LCD display, pro zobrazení okamžité rychlosti pak bude

připraven výrazný display segmentový. K základním údajům, jako je okamžitá rychlost, patří též možnost přehledného zobrazení otáček motoru obstaratelného řadou LED.

Pro volbu zobrazovaných dat bude zobrazovací modul osazen tlačítkem, stejně tak pro reset jednotky a pro spuštění záznamu vybraných dat na přenosné paměťové medium, jehož slotem bude modul též disponovat.

3. Rychloměr 1.0

Z pohledu na výčet požadavků je jasné, že se jedná o poměrně komplexní a nesourodý projekt. Z toho důvodu, během řešení jednotlivých nároků, byla v průběhu diplomové práce sestavena pre-jednotka, zobrazovací zařízení, nazvěme ho generace 1.0. Rychloměr zamýšlený a konstruovaný především k ověření praktické funkčnosti čidel a zkoušení zápisu na přenosné paměťové médium. Tyto budíky se jako rychloměr osvědčili, neustálé problémy se zápisem na kartu pak daly podnět k přezkoumání zapojení a změny pro generaci 2.0. Bohužel tato zpráva nebude obsahovat detailnější fotodokumentaci laboratorní realizace, protože těsně před plánovanou fotodokumentací došlo k nesmyslné, až tristní ztrátě tohoto prototypu. Dochovaná fotografie vozidla s rychloměrem na obr. 3.1. Jednotka rychloměru je bílého kelímkovitého vzhledu, umístěná těsně před volantem.



Obr. 3.1 : Elektromobil s prototypem rychloměru 1.0

3.1 Čidla

První komponenta nesoucí nejasnou specifikaci a potřebu ozkoušení, byla čidla ABS od firmy ATE umístěná u kol v těhlicích snímající magnetické kroužky se čtyřiceti dvěma unipóly (obr. 3.3). První úloha k řešení nastala už ve chvíli připojení čidla k obvodu a to nedostupností poměrně speciálního konektoru užívaného u ABS systémů. Řešením pak byly 2,8 mm fastony naražné na dvou kontaktech, kterými čidlo disponuje, označených na plášti čidla znaménky – a + . Tato znaménka se

nám pak stala návodem pro pokus napájení čidel a měření jejich výstupu při otáčení magnetickým kroužkem v blízkosti snímače. Výsledkem pokusu bylo zjištění



proměnlivého úbytku napětí, při jmenovitém napájení 9VDC z lineárního stabilizátoru 7809 [13] se hladiny, na něž výstup klesal, rovnaly hodnotám 7,26 a 5,7V. Nepísemná část této práce zahrnovala v následujícím kroku rozmístění čidel na vozidle a rozvedení kabelů od čidel, až do konektoru rychloměru.

Obr. 3.3 : Snímač čidla vystupující z těhlice k magnetickému kroužku



Obr. 3.2 : Čidlo v pravé zadní těhlici

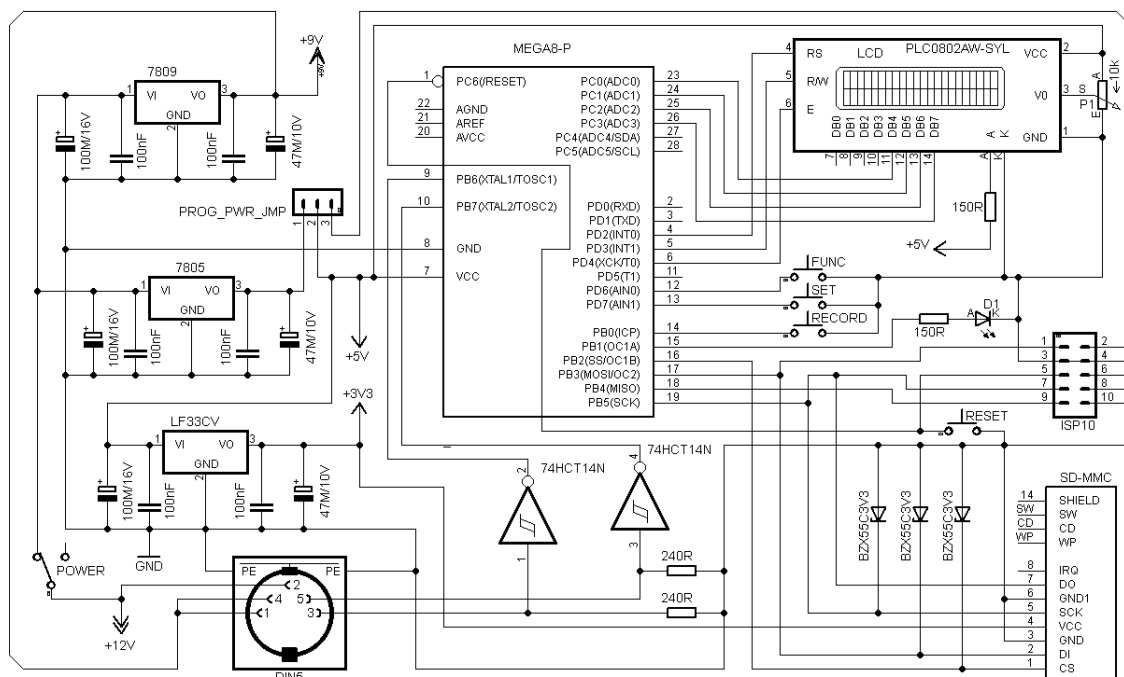
3.2 Schéma rychloměru

Chování čidel ověřené experimentem je dáno, může tedy následovat podkapitola řešící návrh zapojení rychloměru 1.0. Výčet požadavků na osazení rychloměru korespondující s Obr. 3.4, s přílohou A a jejich uspokojení na osazení zde. Rychloměr měl za úkol zaprvé zobrazovat okamžitou rychlost, eventuelně čítat a zapisovat dlouhodobou ujetou vzdálenost (odo) a krátkodobou (dist). Tento požadavek pokryl podsvícený alfanumerický LCD display plc0802aw-syl [14] s dvěma řádky po osmi znacích. Volbu zobrazovaných údajů obstarával mikrosplínač (FUNC).

Srdcem rychloměru se pak pro pokusné účely stal v daný okamžik při ruce dostupný mikrokontrolér Atmel z rodiny AVR, ATmega8 [15]. Tento jedno-čip disponuje především pro nás zajímavými třemi osmibitovými vstupně výstupními porty, SPI rozhraním (Serial Peripheral Interface Bus) pro naši potřebu spoustou paměti (8kB program FLASH, 1K Byte SRAM, 512byte EEPROM). Čip je libovolně programovatelný v jazyce C, ve volném vývojovém prostředí AVRStudio, cenově dostupnými programátory rozhraní ISP10 (vstup na DPS). Ovládání LCD je dle schématu na Obr. 3.4 obsluhováno čtyřmi bity portu C přenášejícími data a třemi řídicími signály portu D. SPI rozhraní je pak využito kromě vstupu programátoru také k připojení přenosného paměťového média SD/MMC karty.

Připojení SD karty dává k řešení problematiku úrovně napájení, neb ATmega8 používáme u našeho prototypu s logikou TTL (tranzistorově-tranzistorová logika) s napájením 5 VDC, vstupy a výstupy karty pak pracují s TTL 3,3V. Abychom kartu nepoškodili, byl prototyp osazen kromě zdroje napájení 5V (v našem případě lineární stabilizátor 7805 [13]) i stabilizátorem LF33CV pro napájení karty 3,3V, 5V logické signály ATmega8 obsluhující kartu pak sníženy zenerovými diodami BZX55C3V3 [16] na požadovanou úroveň 3,3V.

Poslední část obvodu pak řeší připojení čidel. Pár čidel je připojen k rychloměru přes DIN9 konektor, jejichž kontakty označené znaménkem + jsou připojeny k pinům DIN9 konektoru číslo 1 a 4 a výstupní, záporným znaménkem popsané kontakty napojeny na piny 5 a 3 (stínění konektoru a pin 2 jsou využity k napájení rychloměru 10-15VDC). Výstupy čidel jsou připojeny na vstupy Schmittova invertoru 74HCT14N [17], dělené proti zemi přes 240R odpor tak, aby překlápěli výstup Schmittova invertoru a ten nám tak převedl dvě napěťové úrovně čidla na 5V TTL, která následně zpracuje ATmega8 na šestém a sedmém vstupu portu B.



Obr. 3.4 : Schéma zapojení rychloměru generace 1.0

3.3 Výpočet rychlosti a software

Od impulsu k rychlosti. Začneme u kola. Náš elektromobil obouvá pneumatiky o rozměru 195/55 R15. K výpočtu obvodu kola se pak jedná o údaj o šířce pneumatiky 195mm, výška profilu pneumatiky udaná v procentech šíře pneumatiky tj. 55% z 195mm čili 107,25mm a průměr ráfku kola 15 palců = 381 mm. Průměr celého kola je pak roven součtu dvojnásobku profilu a průměru ráfku, v našem případě tedy 595,5 mm. Obvod kruhu je roven $\pi \cdot d$, naše kolo tak na jednu otáčku urazí vzdálenost 1871 mm. Jedna otáčka kola je magnetickým kroužkem čidla rozdělena na 42 rozpoznatelných dílků, jeden dílek po zaokrouhlení na stranu bezpečnou (z bezpečnostního hlediska může rychloměr přeměřovat) má tak 44,5 mm.

Jako vhodná obnovovací frekvence zobrazení rychlosti byla zvolena doba rovná polovině vteřiny. Měříme-li dvě čidla najednou, dostáváme každou 0,5 sec. z jejich součtu počet impulsů rovný dvojnásobku času čili jedné vteřině, zároveň se nám tím údaje čidel průměrují. Výpočet rychlosti je pak otázkou mžiku a to prostým vynásobením počtu impulsů a délky jednoho dílku za čas, tedy rychlost v kilometrech za hodinu je rovna součtu impulsů z obou kol za půl vteřiny násobeno (4,45x3,6) děleno stem. Následné čítání ujeté vzdálenosti se děje po stovkách metrů. Jedno sto metrů je

rovno vzdálenosti 2247 dílků, čítáme-li dvě čidla, pak po každých 4494 impulsích navýšíme ujetou vzdálenost o 100m. Dlouhodobá vzdálenost je ukládána na eeprom paměť ATmega8 a je při zapnutí vždy opětovně vyvolána.

Pro ukázkou ve zdrojovém kódu pak výpočet rychlosti a čítání ujeté vzdálenosti a ukládání na eeprom paměť vypadá asi takto:

```
//obnova displaye a provadeni vypoctu kazdych 500ms;
if (refr > 500) {
    mod = eeprom_read_byte( &mode);
    traillong = traillong + chain1 + chain2;
    if (traillong > 4494) {
        traillong = traillong % 4494;
        vzdalenostinc()
        vzdalenostAllinc();
    }
    rychlost = (chain1 + chain2) *16 / 100;
    refr = 0;
    if (touched > 0 && touched <3) {
        touched--;
    } else {
        ClearScreen();
        SendInt(rychlost, 1, 0);
        if (mod == 1) {
            SendFloatZero(vzdalenostAll, 2, 0);
        } else {
            SendFloat(vzdalenost, 2, 0);
        }
        SendString("km/h", 1, 4);
        SendString("km", 2, 6);
    }
    chain1 = 0;
    chain2 = 0;
}
else {
    refr++;
}
```

Kompletní zdrojový kód včetně rutin programu na přiloženém CD
ve složce SW\1.0 gen\1.0b.

4. Gen 2.0 – hardware

Tato kapitola se zabývá volbou a návrhem vhodného hardwaru řídicí jednotky elektromobilu. V předchozích částech práce byly analyzovány potřeby a nároky elektromobilu na jednotku a všechny periferie k jednotce, ať už přímo, či nepřímo připojené. Tvorbou pre-prototypu jsme získali cenné informace o fungování a nefungování možných návrhů zapojení, ze kterých budeme v následující stranách také vycházet. Nyní se věnujme návrhu zapojení desky plošných spojů a volbě jednotlivých komponent potřebných pro vyhotovení zařízení a jeho správnou funkci.

K řešení jsou hned dva možné pohledy na účelovost jednotky. Celé zařízení musí vhodně ovládat provozní režimy udané spínací skříňkou a zároveň musí zvládat odečítání zobrazování a ukládání dat dle volby uživatele elektromobilu.

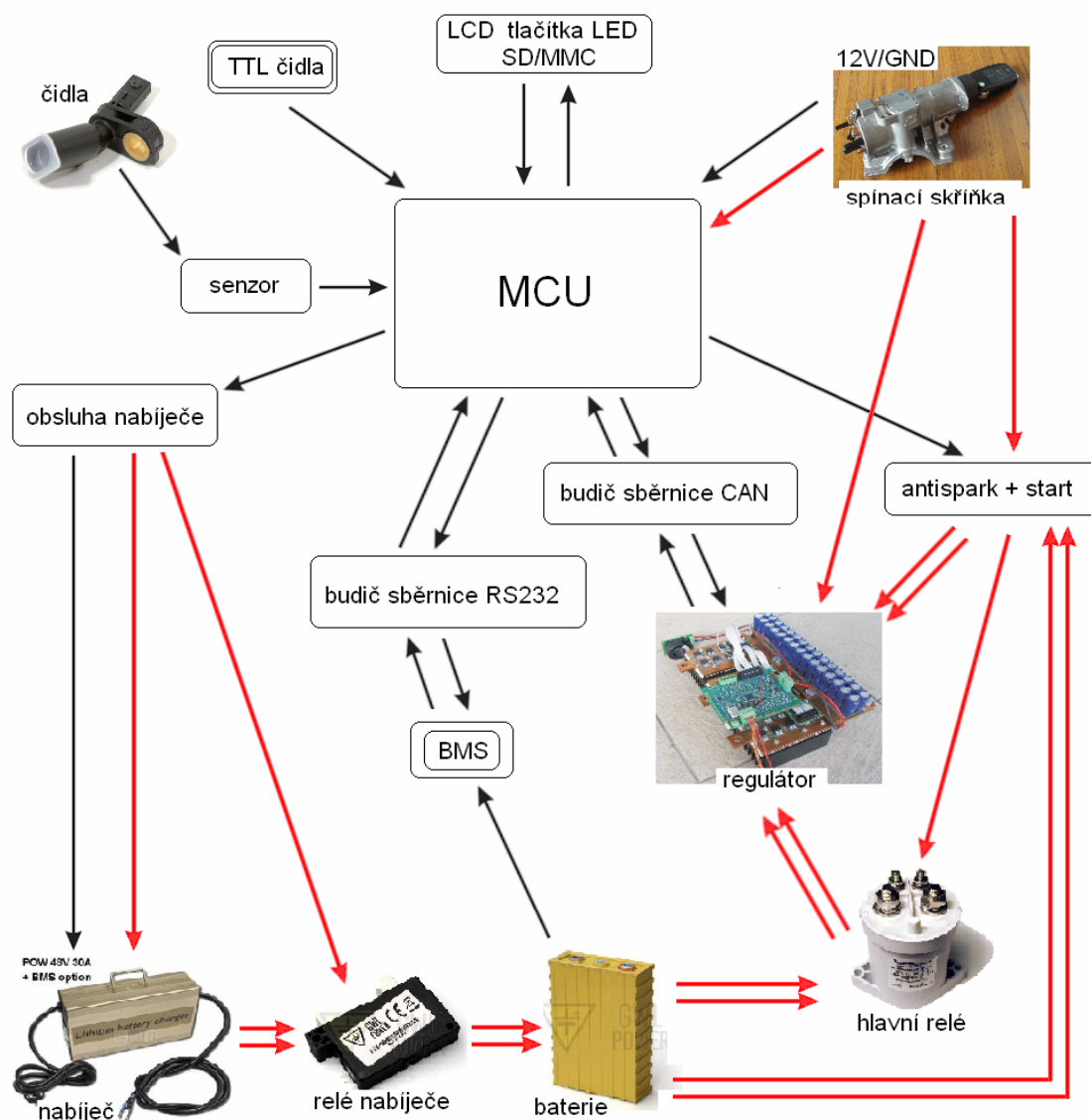
Z hlediska provozních režimů se částečně hardwarově musí počítat s ovládáním nabíječe, zahrnující i odpojení od baterií prostřednictvím relé. Podklady pro softwarovou část řízení nabíječe budou dodávány rozhraním RS232 [18] od externího, baterie balancujícího, modulu. Provozní režimy pak dávají nároky na integraci antispark obvodu regulátoru, od TTL logiky odděleného optočlenem a tranzistorového spínače hlavního relé. Jejich spínání pak bude ponecháno softwarové části práce neboť je možností dalšího odlaďování elektromobilu. K ochraně regulátoru pak je třeba zohlednit potřebu nesoftwarově spínaného zdroje napájení řídicí elektroniky, například přímo ze spínací skříňky, v případě náhlého odpojení od pohonných článků ať už z jakéhokoli důvodu.

Část uživatelská – zobrazující a datová si jednak může brát údaje o stavu baterií z dat pro nabíječ od balancujícího modulu, následně pak také přebírat snímaná data regulátoru prostřednictvím sběrnice CAN. V další řadě musí návrh zapojení pamatovat dostatek vstupů pro snímače otáček kol zkoušených v minulé kapitole, dále pak uvažovat vstupy pro další čidla, například snímače otáčení volantu případně čítač otáček motoru pro vizuální zobrazení. Tím se dostáváme k části zobrazovací. Tato musí být dle požadavků schopna vypisovat i slovně nadepsaná data, zároveň jasně oznamovat údaj o rychlosti a otáčkách motoru. Větší množství typů dat pak lze při zobrazování tlačítkem přepínat a volit mezi nimi. Uživateli též tato jednotka musí dát možnost resetování jednotky v případě chyby. Volič záznamu dat pak společně s přenosným médiem je vhodné navázat v této části.

4.1 Blokové schéma řídicí jednotky

Nyní, když je známo celkové propojení jednotlivých částí elektromobilu a jejich vzájemné vztahy a závislosti, je vhodné vyobrazit si hlavní části, jež musí modul obsahovat. Pro názornost uvedení je zhotoveno blokové schéma řídicí jednotky obr. 4.1, ze kterého bude vycházet další návrh a konstrukce. Schéma lze rozdělit na několik hlavních částí, jež budou popsány ve sledu následujících podkapitol. Do blokového schématu je začleněn tok informací a návaznost jednotlivých modulů.

Popis schématu. Zobrazené vztahy pro porozumění znamenají zhruba toto, dvojité červené šipky znázorňují hlavní pohonné napájení, jednoduché červené šipky pak 12V impulsy spínání a napájení spínacích tranzistorů. Černé šipky pak popisují datové toky ovládací/spínané vazby.



Obr. 4.1 : Blokové schéma

Průchod schématem dle jednotlivých kroků vypadá asi takto. Po zasunutí klíče do spínací skříňky a pootočením do mezipolohy 1.-2. dojde k aktivaci MCU a následně budičů sběrnicí, čidel a zobrazovací části. Vráti-li se klíč do polohy první a nebude vytažen, zůstává vše aktivní a k tomu je aktivována obsluha nabíječe včetně sepnutí relé nabíječe a dle BMS dochází k řízenému nabíjení.

Po otočení klíče do polohy druhé obsluha nabíječe nabíječ vypne, jeho relé je odpojeno. V této poloze dojde sepnutí prostřednictvím spínací skříňky 12V napájení řídicí elektroniky regulátoru, dále pak je přivedeno 12V napájení k nesepnutému tranzistoru hlavního relé a nesepnutému optočlenu spínajícího antispark obvod.

Při přechodu do třetí polohy spínací skříňka dá povel MCU, aby zahájila startovací sekvenci, která se skládá chronologicky ze sepnutí optočlenu spínajícího antispark obvod na dobu, kdy dojde k nabití filtračních kondenzátorů regulátoru (cca 3 vteřiny), a následnému sepnutí hlavního relé, po kterém okamžitě následuje rozepnutí antispark obvodu.

Při přechodu z třetí polohy do druhé pak dojde k rozepnutí hlavního relé, při zachování napájení regulátoru a spínacích tranzistorů, které může být bezpečně odpojeno při přechodu z druhé polohy do první, kdy je opět aktivována obsluha nabíječe. Senzory, TTL čidla a zobrazovací/ovládací část jsou aktivní po celou dobu do vytažení klíče ze zapalování. Údaje z regulátoru jsou přístupné po CAN, až když je klíč otočen do polohy druhé nebo třetí.

4.2 MCU

MCU neboli Multipoint Control Unit má na starosti správné ovládání a řízení celého zařízení, v našem případě, stejně jako dnes již v mnoha dalších složitějších moderních elektronických zařízeních, je srdcem celého obvodu. Při volbě vhodného MCU je nutné se zaměřit na požadavky, jenž jsou na řídicí jednotku kladeny.

Dle čeho vybírat na jaké bázi celý projekt postavíme. Kritéria jsou následující. Dostupnost, odolnost, rozšiřitelnost, spotřeba a pro vývoj dostupnost součástek a čipů ideálně v DIP pouzdrech. Ve vybíraných platformách mezi sebou soupeří tři rodiny mikroprocesorů AVR (Atmel), PIC (Microchip) a ARM (ST). O CPLDS/FPGA uvažovat cenu příliš nemá, neb zásadně pokulhává cenová dostupnost a dostupnost obecně. Při výběru mezi těmito mé rozhodování směřovalo k AVR. ARM platforma sice vyniká vysokým výpočetním výkonem, taktem v řádu desítek MHz a schopností

zpracovat relativně velké toky dat (například z malé kamery), ale pro jednodušší elektroniku obstarávající spínání s napětím v řádu desítek voltů, sběru dat z čidel a zobrazování informací, je platforma převážně pracující na 3,3V TTL nevhodná. Vyřazení platformy PIC z výběru je pak asi otázka vyznání.

Tušení programové náročnosti a potřeby dostatku portů při zachování DIP pouzdra vedlo k výběru jednočipu platformy AVR a to ATmega644-20PU [19] s doplněním rozhraní CAN prostřednictvím Stand-Alone CAN Controller MCP2510-I/P [20]. Ten však vyžadoval obsazení sběrnice SPI, která byla požadavky vyčleněna pro komunikaci s přenosným paměťovým médiem. Proto byl výběr zaměřen na to, aby zvolený MCU obsahoval řadič CAN, tím se i návrh celého zařízení značně zjednoduší. Dále z požadavků vyplývá, aby jednotka obsahovala komunikační linku RS232, čili aby vybraný mikrokontrolér obsahoval řadič UART. Přítomnost ISP pro snadné naprogramování je výhodou. Na výpočetní výkon nenáročná obsluha komponent elektromobilu, sběru a zobrazování vybraných dat pak 8-bitové MCU programovatelné ve vyšším programovacím jazyce C potvrzuje. Souhrnem těchto požadavků, až na nedostupnost v DIP pouzdře, byl zvolen MCU od firmy Atmel AT90CAN 128 [21].

Jak již bylo předesláno, jedná se o 8-bit mikrokontrolér platformy AVR, založené na CMOS technologii. Obsahuje redukovanou instrukční sadu instrukcí RISC (Reduced Instruction Set Computer). Jednočip je založen na harvardské architektuře, má tedy oddělenou paměť dat od paměti programu. Paměť programu je tvořena elektronicky programovatelnou FLASH pamětí o velikosti 128KB. Paměť pro data má pak 4KB a je tvořena statickou polovodičovou pamětí SRAM. Programování FLASH lze realizovat přes rozhraní JTAG, pro mě pak především rozhraním ISP. Výčet základních vlastností AT90CAN128 následován vyobrazením architektury.

• **Přehled vlastností AT90CAN128 [21] :**

• Periferie:

- programovatelný Watchdog s On-chip oscilátorem
- 8-bitový synchronní časovač/čítač (Timer/Counter-0): 10-bitová předdělička, externí Event Counter, výstup Compare nebo 8-bitový PWM výstup
- 8-bitový asynchronní časovač/čítač (Timer/Counter-2): 10-bitová předdělička, externí Event Counter, výstup Compare nebo 8-bitový PWM výstup, 32KHz oscilátor pro RTC operaci
- duální 16-bitové synchronní čítače/časovače: 10-bitová předdělička, vstup Capture s rušením šumu, externí Event Counter, 3 výstupy Compare nebo 16-bitový PWM výstup, modulace výstupu Compare
- 8-kanálový 10-bitový SAR ADC
- Two-wire sériové rozhraní (I2C)
- duální programovatelný sériový USART
- Master/Slave sériové rozhraní SPI
- Řadič CAN 2.0A a 2.0B
- On-chip analogový komparátor

• Speciální vlastnosti MCU:

- Power-on Reset a programovatelná detekce Brown-out
- Interní kalibrovaný RC oscilátor
- 8 externích zdrojů přerušení
- pět režimů spánku: Idle, ADC Noise Reduction, Power-save, Power-down, Standby
- Global Pull-up Disable

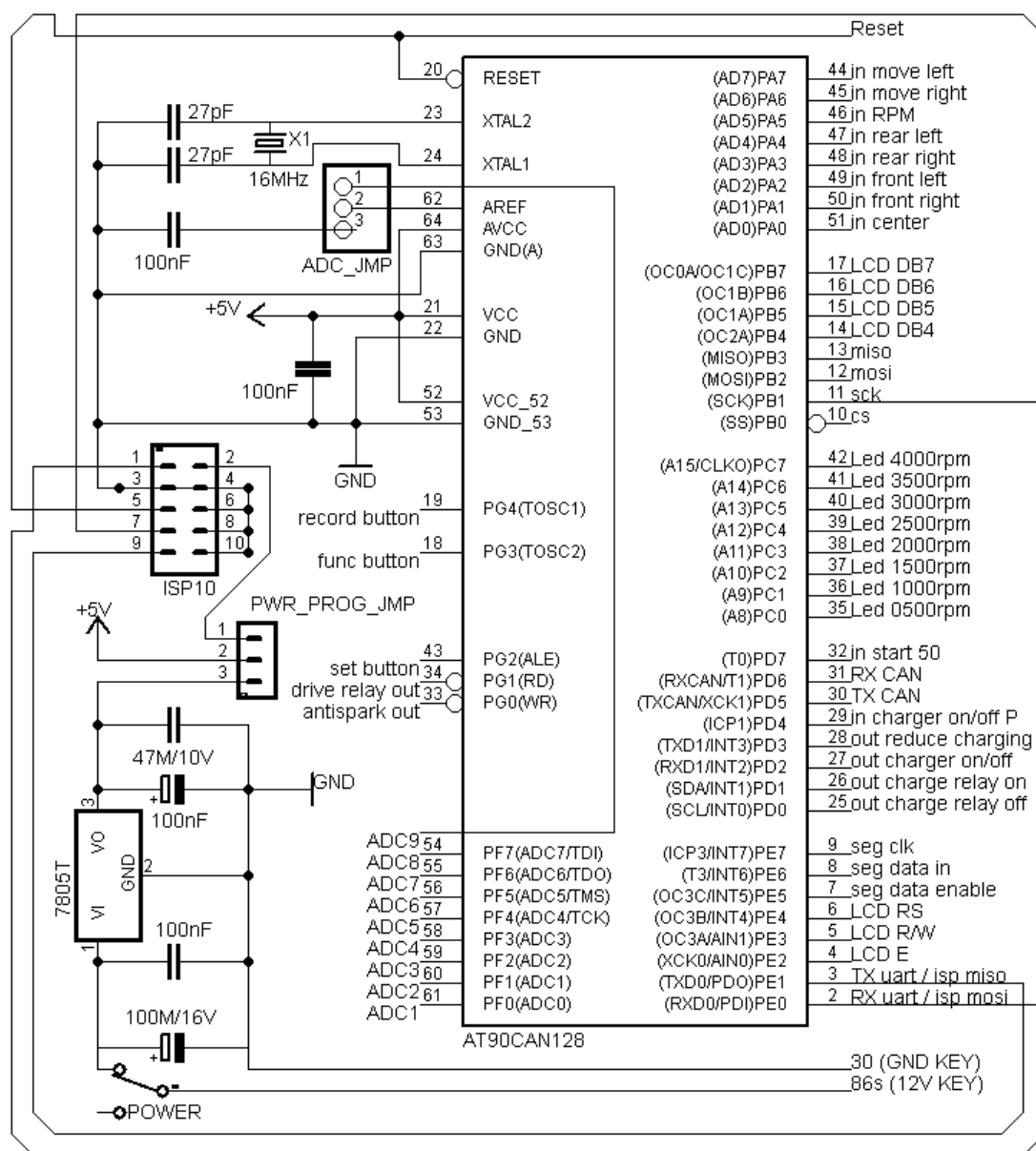
• I/O: 53 programovatelných I/O linek

• Pracovní napětí: 2.7 až 5.5V

• Rozsah teplot -40 až 125°C

• Maximální frekvence: 8MHz při 2.7V, 16MHz při 4.5V

Komplexnější popis je k nalezení v datasheetu výrobce [21] společně s podrobnějším popisem jednotlivých pinů. Na obr 4.3 je popis pouzdra s funkcemi jednotlivých pinů. Naše zapojení mikrokontroléru je pak vyobrazeno na obr. 4.4, piny jsou jednotlivě popsány dle zapojení a popis koresponduje se schémata v dalších podkapitolách. Na obrázku 4.4 jsou mimo MCU zobrazeny další podpůrné součástky potřebné pro funkci. Byť má jednočip AT90CAN128 vlastní interní oscilátor, je dobré se na něj nespolehat a obvod doplnit oscilátorem externím, v našem případě o jmenovité hodnotě 16MHz. Na schématu je pro ilustraci zachycen napájecí obvod lineárního stabilizátoru 7805 [13], zapojení programovacího konektoru ISP10 a propojka pro volbu napájení z programátoru / stabilizátoru 7805.



Obr. 4.4: Schéma zapojení MCU v řídicí jednotce

Rozvedení pojmenování pinů na portech je následovné. Port A je vyhrazen na pinech PA1 až PA4 pro vstupy snímačů otáčení kol v pořadí PA1- pravé přední, PA2 – levé přední, PA3 – pravé zadní a PA4 - levé zadní, PA0 je pak určeno k informaci o střední poloze řízení (před mechanickou úpravou řízení prozatím simulováno tlačítkem). Pin portu A číslo 5 předává signál od Hallovy sondy A1302 [22] snímající otáčky motoru, PA6 a PA7 jsou signály od dvou Hallových sond TLE4905G [23] snímající otáčení magnetického kroužku na tyči řízení. Horní 4 bity portu B jsou vyhrazeny pro obsluhu datových vstupů LCD displeje, spodní 4 bity pak umožňují prostřednictvím SPI rozhraní komunikovat s přenosnou pamětí typu SD/MMC.

Port C je celý obsazen spínáním osmi dvojic barevně odlišených LED diod indikujícím otáčky elektromotoru motoru. Rozhodnutí obsadit celý port LED bylo sice neúsporné z hlediska využití vstupů, ukázalo se však velice praktické z hlediska následného odlaďování programu.

Při pohledu na port D zaregistrujeme využití speciálních funkcí dvou pinů, a to PD5 a PD6, je k nim přiveden budič sběrnice CAN MCP2551-I/P [24]. Vývod PD7 slouží k příjmu vstupní informace ze spínací skříňky hlásící sepnutí třetí polohy klíče a tedy spuštění antispark/startovací sekvence. Bity PD0 až PD4 jsou určeny pro první polohu klíče a starají se o obsluhu nabíječe. Jmenovitě PD4 přijímá přes optočlen vstupní informaci o návratu klíče do první polohy a aktivuje obsluhu nabíječe. PD2 a PD3 pak řídí obsluhu režimy nabíjení. PD0 a PD1 jsou určeny k přepínání bistabilního relé mezi nabíječem a baterií.

Popis portu E zahájím využitích speciálních funkcí na PE0 a PE1, tedy sběrnice UART a jejich napojení na IO MAX232 [25]. Tyto dva vývody též slouží jako vstup a výstup dat ISP rozhraní při programování, toliko rozdílné zapojení oproti rodině AVR ATmega, což zadalo z počátku snahy o naprogramování zadalo několika chvilím nervozity a dohledávání tohoto správného zapojení. Piny 2 až 4 jsou na portu E určeny pro řídicí signály alfanumerického LCD displeje, vývody 5 až 7 pak obsluhují trojmístný segmentový LED displej s integrovaným ovladačem. Port F se speciální funkcí AD převodníků je připraven pro libovolné aplikování ať už digitálních vstupů a výstupů a pro možnost potřeby nespécifikovaného měření. K poslednímu portu G je na pinu č. 2 připojeno tlačítko s pojmenováním set, k třetímu pinu tlačítko sloužící k přepínání dat zobrazovaný na LCD, uzemnění pinu D je přiřazeno počátek a ukončení záznamu dat na přenosnou paměť. První a druhý bit portu G pak obsluhují sepnutí startovací sekvence třetí polohy spínací skříňky.

4.3 Obsluha nabíječe

O nabíječi POW48V30A/BMS (obr. 4.5) již byla v minulých kapitolách trocha informací přednesena, nyní je z manuálu [10] od prodejce doplníme. Nabíječ je napájen ze sítě 230V. Dle specifikací datasheetu je jeho nominální nabíjecí napětí 48V s nabíjecím proudem 30A, při rychlém nabíjení 64V, což odpovídá provozu alespoň 16-ti článků LiFeYPO4. Nabíječ je vybaven napěťovou ochranou 140V a proudovou 35A. Pomalé blikání indikační LED diody značí redukované nabíjení, rychlé blikání indikuje režim rychlého nabíjení. Kompletní nabití signalizuje rozsvícení



Obr. 4.5 : Nabíječ [10]

LED. Nejzajímavější část manuálu se pak věnuje ovládání BMS konektoru.

Pro přehlednost znázorněno na obr. 4.6 tabulkově. Na obr. 4.7 schématické zapojení vstupů a výstupů nabíječe.

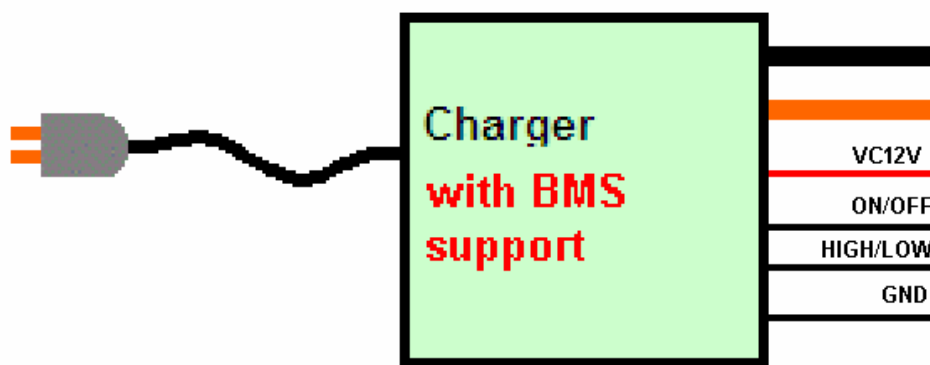
| Contacts | Status | Comment |
|----------|---------------|---|
| VC12V | not connected | the charger works the standard way at MAX power |
| ON/OFF | not connected | |
| HIGH/LOW | not connected | |
| GND | not connected | |

| Contacts | Status | Comment |
|----------|--------|---|
| VC12V | +12V | the charger works the standard way at MAX power |
| ON/OFF | GND | |
| HIGH/LOW | GND | |
| GND | GND | |

| Contacts | Status | Comment |
|----------|---------------|---------------------------|
| VC12V | +12V | the charger stops working |
| ON/OFF | not connected | |
| HIGH/LOW | any | |
| GND | GND | |

| Contacts | Status | Comment |
|----------|---------------|---|
| VC12V | +12V | the charger works the standard way at REDUCED power (10%) |
| ON/OFF | GND | |
| HIGH/LOW | not connected | |
| GND | GND | |

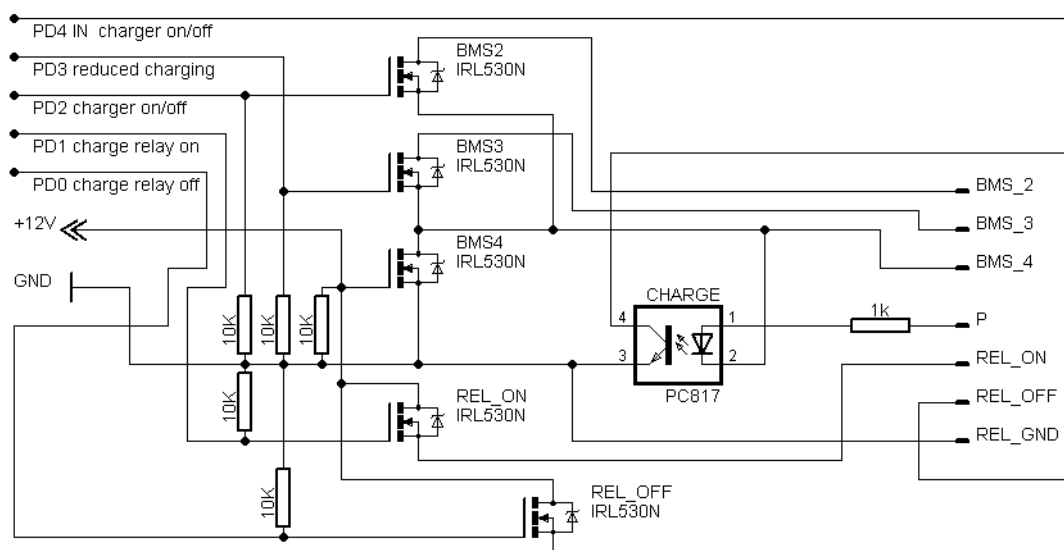
Obr. 4.6 : Variace zapojení BMS konektoru nabíječe [10]



Obr. 4.7 : Zapojení BMS konektoru nabíječe [10]

K čemu vlastně BMS neboli Battery Management System slouží? Celá baterie pro pohonné ústrojí elektromobilu se skládá z více článků. I když jsou takové články takřka totožné a vykazují na první pohled stejné charakteristiky, při nabíjení v sérii se projevují i sebemenší rozdíly v tom, kdy je který článek dříve nabitý. BMS systém pak slouží k identifikaci nabití jednotlivých článků a jejich ochraně proti přebití. (pokročilejší systémy sledují i jejich rovnoměrné vybíjení). Zároveň takový systém může ovládat nabíječ, určit kdy jsou všechny články správně nabité a podle toho nabíjení vypnout, redukovat či opětovně zapnout čili balancovat.

Pro takové to ovládání nabíječe na základě dat z BMS systému je součástí návrhu a následné realizace naší jednotky zapojení na obr. 4.8.

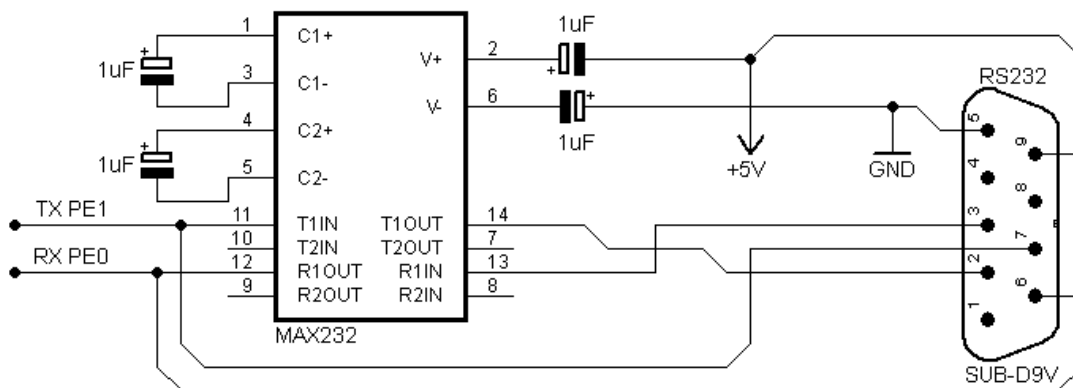


Obr. 4.8 : Schéma zapojení obsluhy BMS konektoru nabíječe

Dle obr. 4.6 slouží první kontakt konektoru k napájení řídicí elektroniky nabíječe, čtvrtý pak jako uzavření napájecího okruhu. Ty jsou v našem zapojení reprezentovány zdrojem 12V shodným s napájením celé naší řídicí jednotky a tranzistorem BMS4, jenž přivede na 4. kontakt zem. Je-li tedy naše řídicí jednotka aktivní (všechny 3 polohy klíče), je aktivní i řídicí elektronika nabíječe, deaktivuje se též společně s vytažením klíče ze spínací skříňky. BMS4 zároveň slouží jako zem pro řídicí tranzistory BMS2 a BMS3 obsluhující spínání druhého a třetího kontaktu BMS konektoru. Důmyslnost vyobrazeného zapojení spočívá ve vstupu P vedoucího v první poloze i v případě vytažení klíče stále potenciál 12V. Tímto vstupem indikujeme první polohu, neb je v dalších polohách odpojen. Aby ale nedocházelo k stálé aktivaci optočlenu CHARGE, převádějícího sepnutí 12V na 5V vstup pro AT90CAN128, je jeho vstup též uzemněn přes tranzistor BMS4, a po vytažení klíče je odpojen. Dle aktivace vstupu P (PD4) je pak řízení nabíjení aktivováno/deaktivováno a bistabilní relé [11] tranzistory REL_OFF a REL_ON rozepnuto nebo sepnuto.

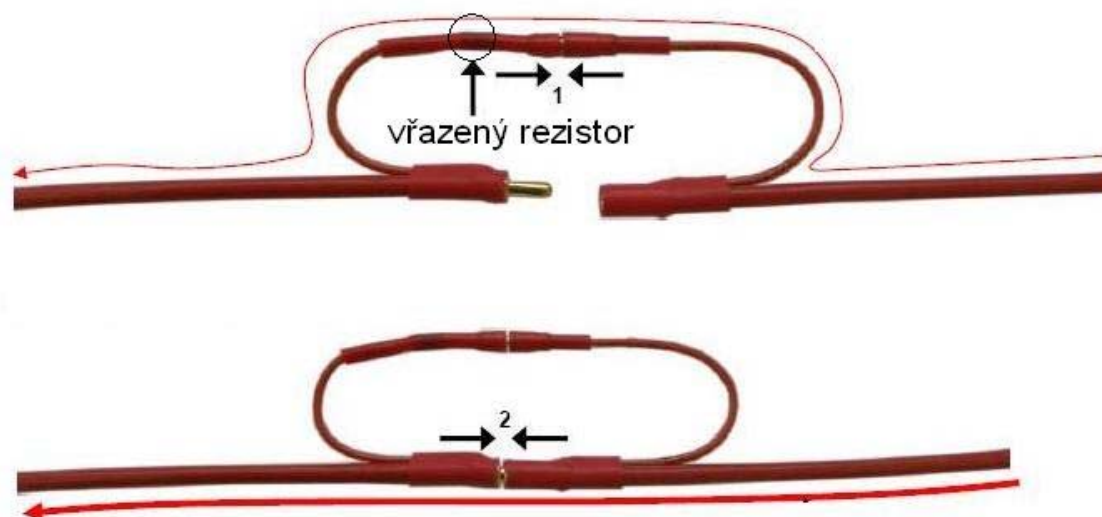
4.4 RS-232

Standard RS-232 [18], přesněji řečeno poslední varianta RS-232C z roku 1969, nebo-li také sériový port, nebo sériová linka se používá jako komunikační rozhraní osobních počítačů a další elektroniky. Pro naše potřeby ho definuje především potřeba komunikace s BMS systémem, jeho přítomnost je však zásadní výhodou pro možnost ověřování funkce ostatních sběrnicí a ostatních částí jednotky. Taktéž lze přes sériovou komunikaci sbíraná data odesílat rovnou do počítače k dalšímu zpracování. Část řídicí jednotky řešící připojení sériové linky je vyobrazena na obr. 4.9. Zapojení je nestandardně rozšířeno o přímý vývod UART na konektor pro možné potřeby BMS.



Obr. 4.9 : Zapojení sériové linky s IO MAX232

Proč startovací sekvence a co to je antispark? Střídavé regulátory, pracující s vícečlánkovými bateriemi a vysokým napětím, jsou při zapojování pohonných akumulátorů sužovány jevem proskakující jiskry při spojování vodičů, z důvodu nabíjení filtračních kondenzátorů regulátoru. Přeskakující jiskra zapojujované konektory a stykače opaluje a ničí. Jev se dá odstranit použitím takzvaného Anti-Spark pomocného obvodu. Pro lepší představu obr. 4.11. Jedná se o předzapojení napájecího vodiče s vřazeným rezistorem (1). Vřazený odpor, volený tak, aby bylo nabití co nejkratší s ohledem na možné zahřátí, zabrání při zapojování pomocného obvodu přeskočení jiskry. Filtrační kondenzátory regulátoru se nabijí a hlavní napájecí vodič je připraven k zapojení již bez přeskočení jiskry.



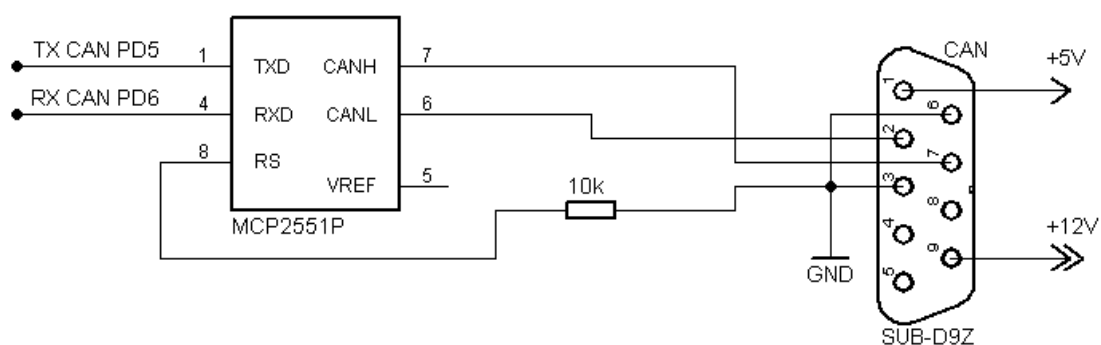
Obr. 4.11 : Principiální znázornění Anti-Spark obvodu

4.6 CAN

Jelikož máme k dispozici regulátor komunikující po CAN, základním parametrem pro návrh jednotky byla schopnost komunikace po této sběrnici. Pro naše účely prozatím stačí komunikace jednosměrná. V rámci možností rozšiřitelnosti elektrických a elektromechanických komponent na elektromobilu, se může dispozice této sběrnice ukázat budoucně jako velice praktická.

Datová komunikační síť s názvem CAN (Controller Area Network) [12] byla navržena koncem 80.let německou Firmou Robert Bosch GmbH pro vlastní potřeby. Záměrem návrhu byla úspora kabeláže a bezpečný přenos informací mezi snímacími řídícími a výkonnými prvky v automobilech. Z této oblasti se však CAN rychle rozšířil díky vysoké rychlosti přenosu, vysoké spolehlivosti a odolnosti při extrémních podmínkách (teplota, rušení apod.) do ostatních průmyslových odvětví.

Elektrické parametry fyzického přenosu jsou specifikované normou ISO 11898 a maximální teoretická rychlost přenosu sběrnice je 1 Mb/s. Tato sběrnice umožňuje výměnu dat mezi řídicími jednotkami a to pomocí datových rámců. Ty se skládají z datových polí, které jsou složené z různě definovaného počtu bitů (1 nebo 0). Pole s datovými informacemi obsahuje vlastní zprávu. Datový rámec obsahuje také pole pro řídicí a kontrolní informace. Komunikace probíhá následujícím způsobem. Jedna z řídicích jednotek vysílá do sítě data. Zbylé jednotky data přijímají a přijatá data vyhodnocují. Pokud shledá řídicí jednotka, že přijaté data jsou určena pro ni, tak je zpracuje, v jiném případě data ignoruje.

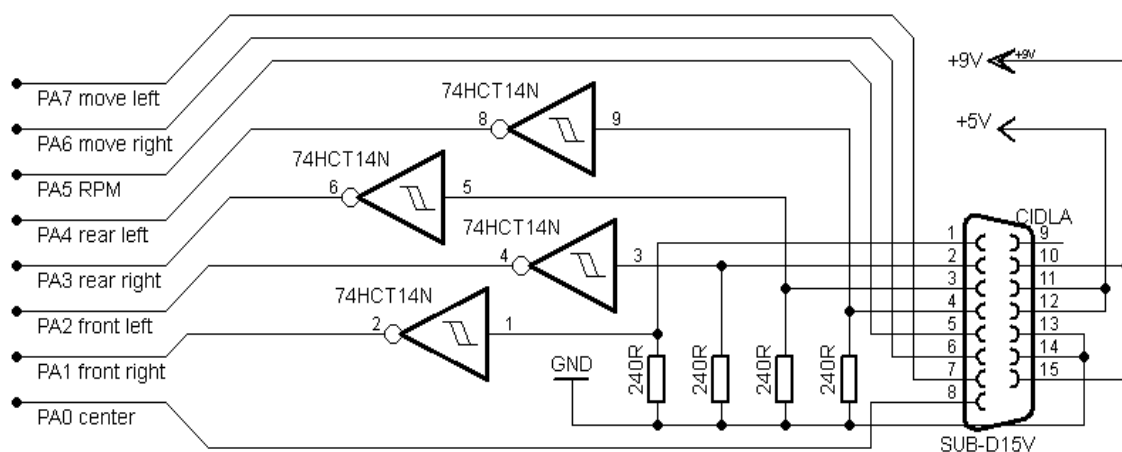


Obr. 4.12 : Schéma zapojení CAN

Fyzické Propojení MCU (obr. 4.12) zajišťuje budič sběrnice CAN. Ten dostává od MCU data jež se mají odeslat po sběrnici a zároveň předává data ze sběrnice do MCU. V našem zapojení je zvolen budič MCP2551 [24] z produkce výrobce Microchip. K zapojení sběrnice je navržen konektor CANNON 9 Female.

4.7 Senzory

Oproti první generaci rychloměru byl senzorický systém na obr. 4.13 rozšířen. Zapojena byla i další 2 čidla snímání otáček kol, s přiřazením k číslům portu A, a to následovně: 1 = pravé přední, 2 = levé přední, 3 = pravé zadní a 4 = levé zadní kolo. Zapojení čidel je dle schématu prakticky totožné jako u první generace (kapitola 3.2). 9VDC je k čidlům přiváděno z pátého a desátého pinu konektoru SUB-D15V, výstupy čidel s dvěma výstupními napětíovými úrovněmi 5,70V a 7,26 jsou přiváděny, po dělení napětí proti zemi přes 240R odpor, ke vstupům Schmittova invertoru 74HCT14N [17]. Dále jsou impulzy v podobě TTL logiky na výstupu Schmittova invertoru předávány ke zpracování mikrokontroléru.



Obr. 4.13 : Senzorický systém

Kontakt č.5 konektoru CIDLA je napojen na výstupní signál Hallové sondy A1302 [22], umístěné u elektromotoru (obr. 4.14), buzené magnetickým polem magnetu, umístěného na ose mezi motorem a převodovkou. Z tohoto signálu je každou polovinu vteřiny vypočten údaj okamžitých otáček a indikován příslušným počtem LED na budíkách.

Další 3 čidla vstupující signálem do jednotky na pinech 6, 7 a 8 konektoru CIDLA, jsou zamýšlena k určení polohy volantu, pro volitelné výpočty správného otáčení kol, a zjišťování nedochází-li k protáčení kol. Tyto údaje jsou použitelné, prostřednictvím sběrnic (například CAN), dále přenositelné k brzdovým systémům, které snad budou na elektromobil aplikovány v budoucnu. Čidlo, snímající a čítající pohyb volantem vlevo a vpravo, je sestaveno ze dvou unipolárních Hallových sond TLE 4905G [23] a převzatého magnetického kroužku z nábojů kol antiblokačního systému. Pro určení střední polohy volantu je zapotřebí čidlo třetí. Mechanicky náročné umístění

do hřebenu řízení se prozatím obešlo napojením na mikrospínač ovládaný uživatelem. Napájení 5V všech Hallových sond je prováděno z pinů 11 a 12 konektoru CIDLA, uzemnění pak na pinech 14 a 13.



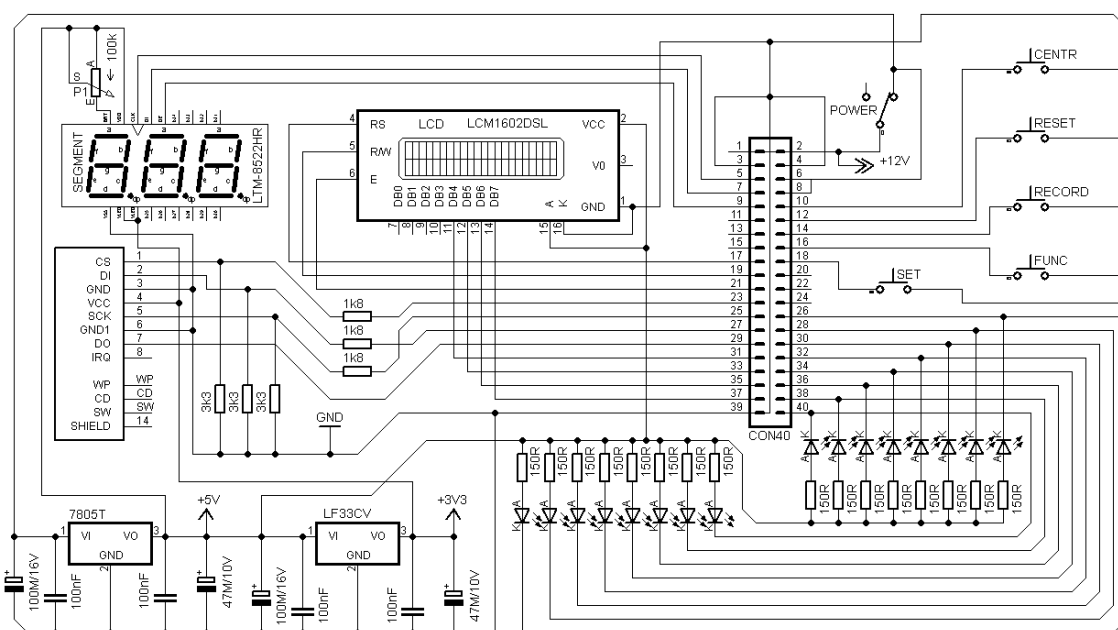
Obr. 4.14 : Snímač otáček elektromotoru

4.8 Zobrazovací modul

Při návrhu této části je třeba pamatovat na vše, co by se mohlo z pohledu jak uživatele, tak tvůrce programu hodit zobrazovat a volit tlačítky. Jakožto automobilistický nadšenec mám jasné dané požadavky na zobrazované údaje. Mezi dvojicí hlavních informací patří zobrazení okamžité rychlosti a alespoň náznakem grafické zobrazení otáček motoru. Na obr. 4.15, v plné velikosti v příloze B, jsou ve schématu vyobrazeny komponenty užité k sestavení zobrazovacího panelu, chcete-li budíků.

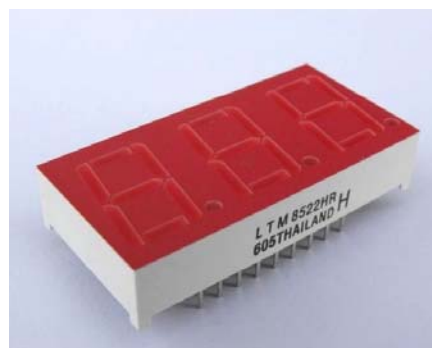
Tato část je pro prototyp navrhována a uvažována jako autonomní modul, oddělený od části s MCU obsluhující sběrnice, sběr dat a řízení. Je tomu tak z důvodu nedostatku prostoru v místě vhodné lokace zobrazovacího zařízení, tj. před volantem řidiče, kde konstrukce umístila příčný nosník znemožňující tvorbu obsáhlejší jednotky s dostatkem konektorů. Zároveň jedinou možností, jak umístit zobrazovač jízdních dat viditelně ve všech polohách volantu, je úchyt přímo k pohyblivému uložení volantové tyče. Proto při pohledu na schéma naleznete netradiční dvouřadý čtyřiceti pinový konektor. Jedná se o spojení s výpočetní částí řídicí jednotky, propojení pak obstarává plochý, čtyřiceti žilový kabel, v našem případě recyklovaný ATA kabel. Tento spoj není

sběrnici, jedná se pouze o prodloužení vývodů řídicí jednotky, na vzdálenost nezbytně nutnou. Přes toto prodloužení jsou obsluhovány všechny zobrazovací periférie, přenosné paměťové medium a několik tlačítek. Zároveň přes tento spoj vede smyčka napájecího okruhu pro centrální část jednotky, vstupující prvním a druhým pinem. Následně se vrací piny 6 a 8, zde ji lze spínačem POWER přerušit a celou jednotku tím deaktivovat. Tento modul má jednotlivé komponenty napájeny z vlastních lineárních stabilizátorů, mimo napájení centrální části.



Obr. 4.15 : Schéma modulu budíků

V návaznosti na rozsáhlé schéma centrální části jednotky obsažené v příloze C, je na kontaktech 26, 28, 30, 32, 34, 36, 38 a 40 přiveden port C, obsluhující svit dvou osmic LED diod, znázorňující okamžité otáčky motoru. Dle maxima otáčení motoru, které je kolem 4000 otáček za minutu, je zobrazovací krok volen po 500 otáčkách. Pásmo otáček 500-1500 je zobrazováno prvními třemi dvojicemi LED zelené barvy, na vývodech 26, 28 a 30, prvními třemi bity portu C. Další tři dvojice LED, náležící pásmu otáček 2000 - 3000, na vývodech CON40 - 32, 34, 36, je volena žluté barvy, obsluhována bity portu C, s čísly 3, 4, 5. Poslední dvě dvojice udávají pásmo 3500 a 4000 otáček za minutu. Jsou zemněny sedmým a osmým bitem

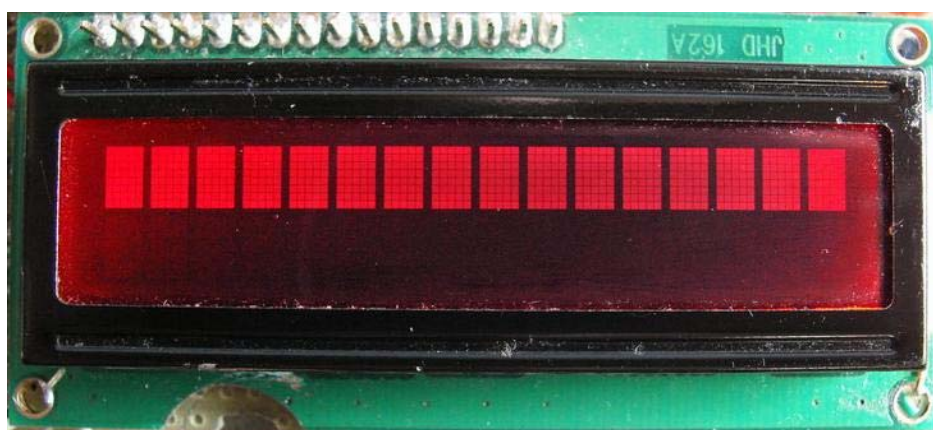


Obr. 4.16 : LTM-8522HR

portu C a mají barvu červenou. Návrh uvažuje obyčejné barevné LED, určené pro napětí 1,5-2V s odběrem 20mA. Pro napájení 5V má každá vřazený odpor 150R. Druhý, důležitý zobrazovaný údaj o okamžité rychlosti, je svěřen trojmístnému 14 mm sedmi segmentovému LED displeji LTM-8522HR [27] obr. 4.16. Tento konkrétní model od výrobce Lite-on electronics, byl vybrán záměrně pro svůj vestavěný integrovaný obvod s posuným registrem M5450. Díky built-in ovladači tak MCU komunikuje s displejem sériově, za využití jednoho datového (CON40 - 7 / PE6), jednoho časovacího (CON40 - 5 / PE7) a jednoho enable signálu (CON40 - 9 / PE5). Displej se dodává pouze s červeným podsvícením.

Osazení zobrazovacího modulu tlačítka je zdůvodněno nejbližší polohou v řidičově dosahu. Pojmenování tlačítek koresponduje se popisem v kapitole MCU a do centrální části jsou přiváděny piny CON40 - 10, 12, 14, 16 a 18. Tlačítko RESET plní funkci resetu jednotky, tlačítko RECORD je určeno pro spuštění a ukončení záznamu dat na přenosnou paměť, SET je prozatím tlačítko volitelné a slouží k dalšímu odladování programu, CENTER simuluje středící čidlo pro snímání úhlu natočení kol a spínač FUNC volí který údaj chceme na alfanumerickém LCD zobrazovat.

Předposlední komponentou je tedy předeslaný alfanumerický LCD displej. Přítomnost tohoto zobrazovacího zařízení je dána potřebou zobrazovat vícero číselných údajů se slovním popisem. Pro zachování jednotnějšího vzhledu jednotky byl vybrán červeně podsvícený, alfanumerický LCD JHD162A [28]. Vzhled zachycen na obr. 4.17. LCD je čínské produkce, s červeným písmem a černým pozadím.

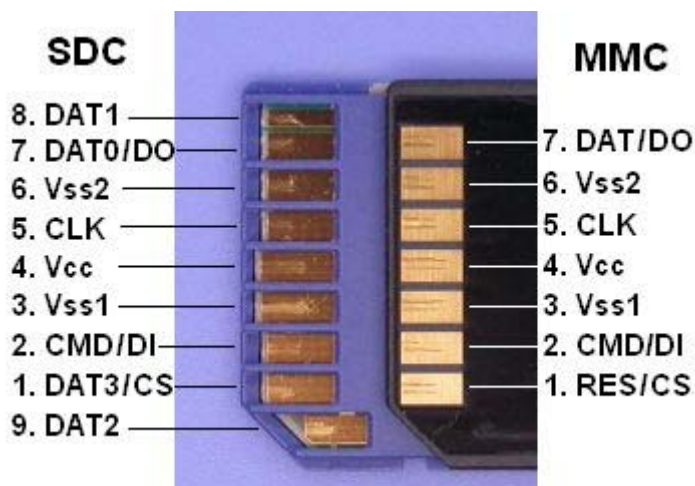


Obr. 4.17 : LCD JHD162A

Displej má dva řádky po šestnácti znacích. Ovládání je kompatibilní s driversy pro řadiče LCD displejů typu HD44780, zapojen je ve čtyřbitovém režimu se třemi řídicími signály. Čtveřici datových vstupů DB4 až DB7 obsluhují čtyři horní bity portu B

(PB4 až PB7), přes piny CON40 -31, 33, 35 a 37. Řídící signály Register Select, Read/Write a Enable jsou dopravovány od MCU přes CON40 - 17, 19, 21.

Poslední periferií, kterou je modul budíků osazen, je slot pro SD/MMC kartu. Zde lze nalézt rozdíl v zapojení oproti rychloměru 1.0. Napájení 3,3V stabilizátorem LF33CV [29] je shodné a připojení komunikačního signálu rozhraní SPI, ve směru karta → MCU, zůstává hardwarově taktéž bez změny. Změna ale nastává v zapojení komunikačních signálů putujících ve směru MCU → karta. Vzhledem k lišící se 3,3V logice karty a 5V logice MCU je, jak bylo uvedeno v třetí kapitole, potřeba snížit 5V hladinu signálů z AT90CAN128 na úroveň 3,3V. V první generaci rychloměru tak bylo činěno prostřednictvím přepadu přes zenerovy diody. Toto zapojení se však neosvědčilo. Zápis na kartu s FAT byl nefunkční. Proto bylo u druhé generace zvoleno odlišného zapojení. Snížení napěťové hladiny signálu je realizováno prostřednictvím odporového děliče s vřazeným rezistorem 1,8k a 3,3k uzemňujícím. Toto zapojení je bezproblémové a spolehlivé. Pro přehledný popis zapojení jednotlivých pinů SD/MMC karet je zde obr. 4.18.



Obr. 4.18 : Zapojení pinů SD/MMC rozhraní [30]

Pro komunikaci s kartou je ke slotu těchto paměťových karet následně přivedena přes CON40 – 23, 25, 27, 29 přivedena sběrnice SPI v pořadí SS (CS), SCK (CKL), MOSI (DI) a MISO(DO).

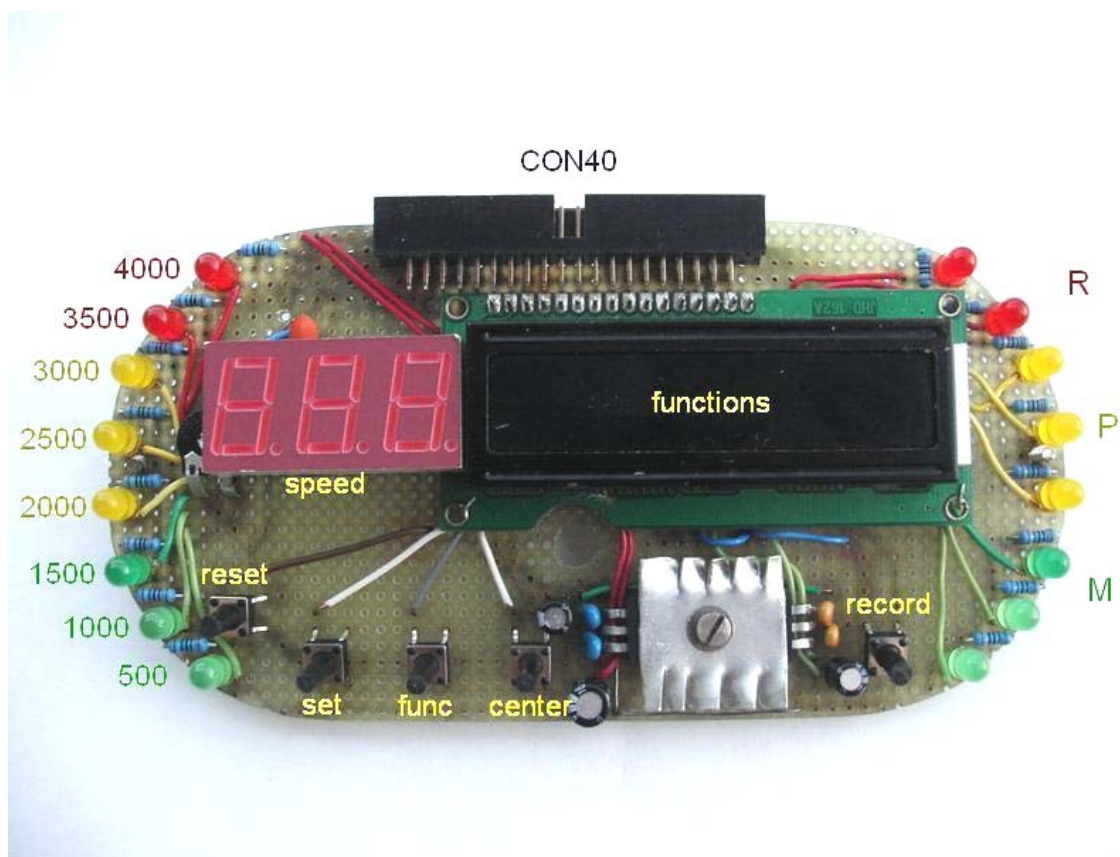
4.9 Realizace

V první části této podkapitoly bude zdokumentováno a popsáno zapojení prototypu řídicí jednotky a hlavně pak jejích vstupů. Zároveň to budiž uživatelským návodem při zapojování na vozidle. V druhé části pak bude popis a fotodokumentace úprav a zapojení spínací skříňky.

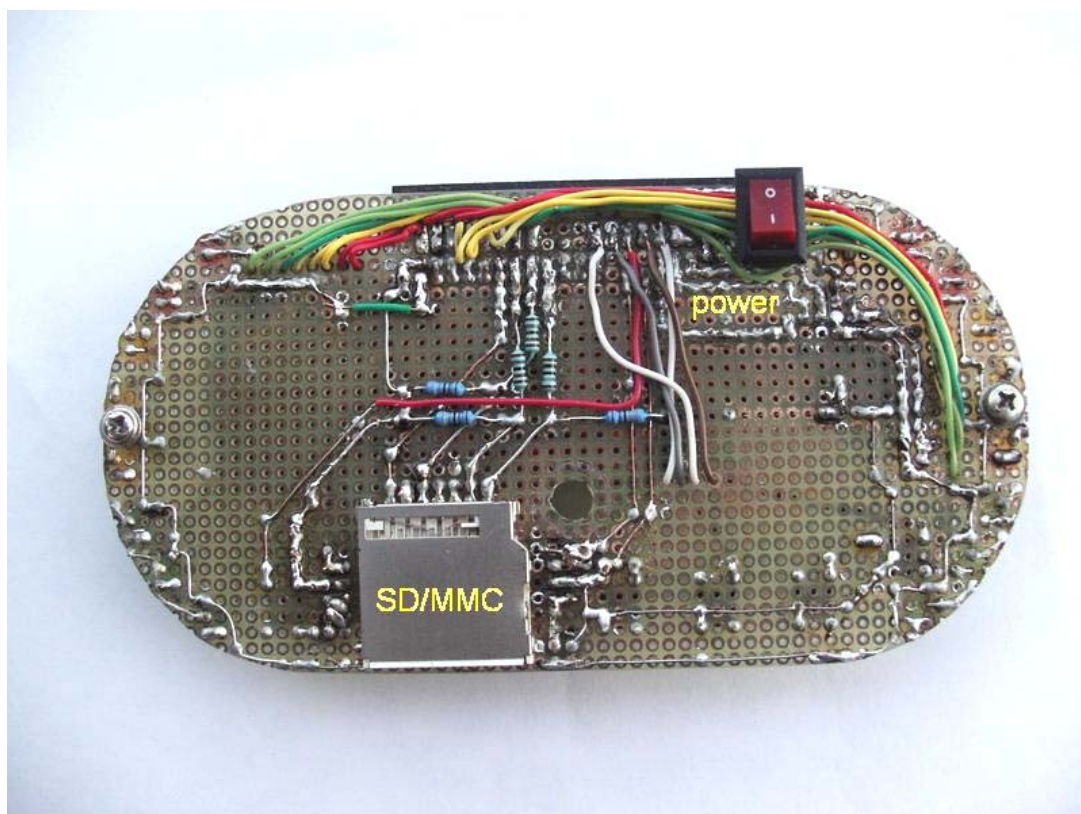
4.9.1 Prototyp

Kompletní dvoudílné schéma zapojení prototypu je vyobrazeno jako část zobrazovací v příloze B, a jako část centrální v příloze C. Realizace prototypu má formu taktéž dvoudílnou, z důvodu nedostatku prostoru v místě vhodné lokace zobrazovacího zařízení. Centrální část jednotky je tak předsunuta a umístěna napevno mezi příčkami držáku řízení, propojena plochým kabelem se zobrazovacím modulem, připevněným na výškově nastavitelném uložení tyče řízení. Prototyp je zapojován na laboratorní desku plošných spojů bez návrhu layoutu, pouze dle schématu .

Popis zobrazovacího modulu bude nejlépe začít obrázky 4.19 a 4.20. Na prvním obrázku je fotodokumentována přední strana desky zobrazovacího modulu, s popisky ovládacích a zobrazovacích prvků. Druhý obrázek ukazuje laboratorní řešení zadní strany desky, vyznačení spínače napájení jednotky a slotu pro SD/MMC kartu.



Obr. 4.19 : Laboratorní prototyp zobrazovacího modulu – přední strana

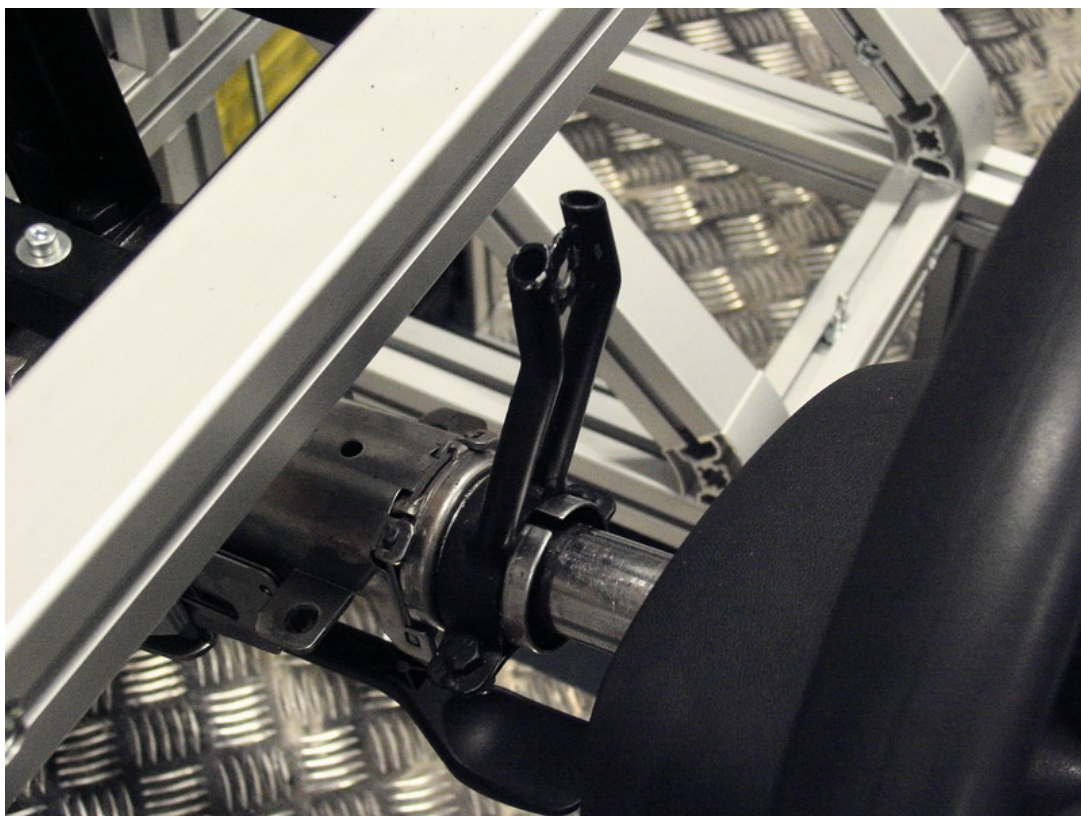


Obr. 4.20 : Laboratorní prototyp zobrazovacího modulu – zadní strana

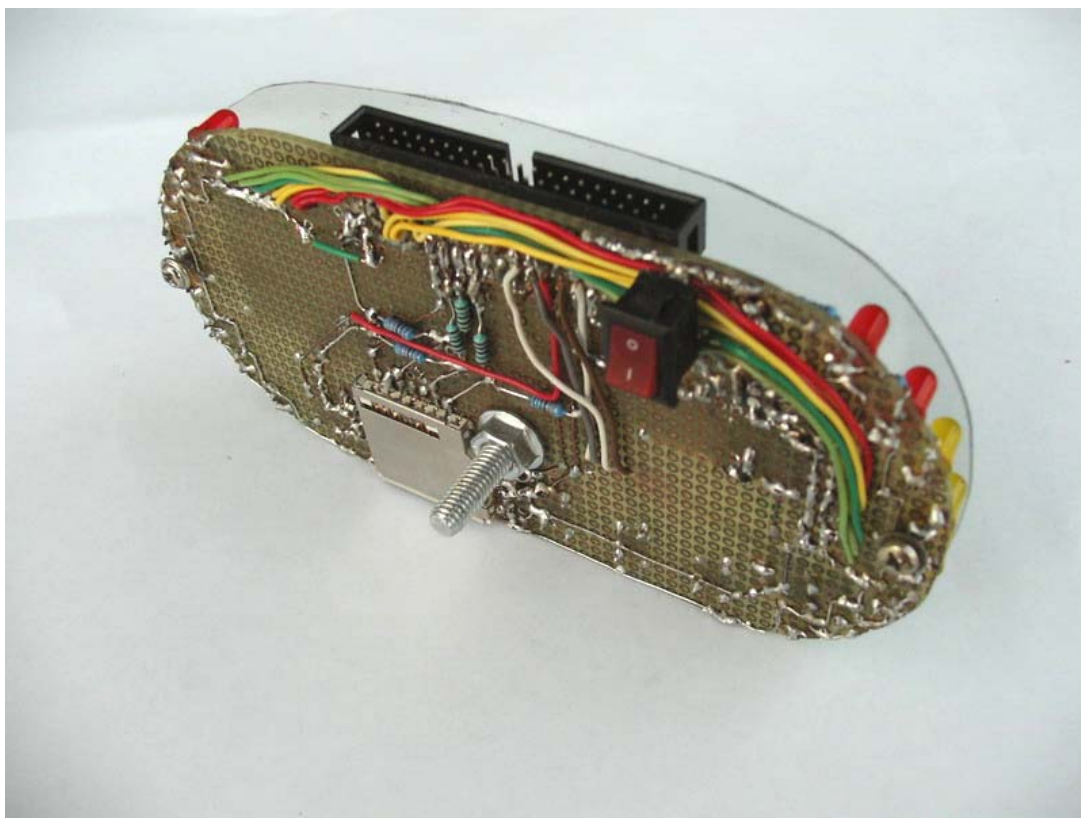
Z fotografií je patrné zapojení součástek dle schématu, v předu rezistory upravující napájení LED diod. Hliníkový chladič ukrývá dva lineární stabilizátory 5V pro napájení LED diod a displejů, 3,3V pro napájení SD/MMC karty a podsvícení segmentového LED displeje. Dále si můžeme všimnout trimru pro nastavení kontrastu segmentového displeje. Na zadní straně desky jsou rozpoznatelné tři dvojice rezistorů napěťových děličů signálů komunikace karty.

Při prvním pohledu na výše zobrazenou desku se podsouvá myšlenka, jak získala svůj tvar. Odpověď je pak skryta v celkové sestavě zobrazovacího modulu a jeho uložení a upevnění na elektromobilu. Již při zkoušení rychloměru 1.0, byl v rámci diplomové práce zhotoven přetvořením trochy železa držák pro zobrazovací zařízení (obr. 4.21). Je vyroben a umístěn na míru tak, aby zobrazovací zařízení nepřekáželo otáčení volantu a zároveň aby bylo možné volant i nadále polohovat.

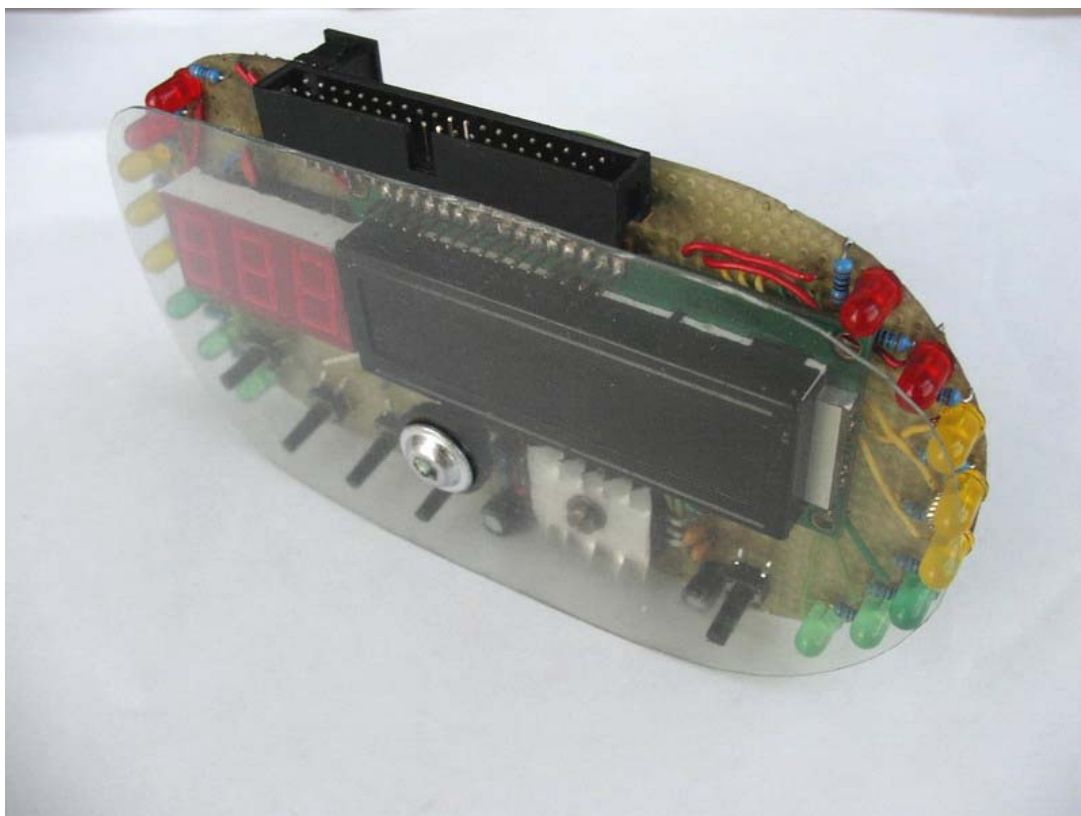
Sestavení budíků je vymyšleno prostým zvrstvením krytů a desky na centrální příchytňý šroub M6 (obr. 4.22), procházející od předního plastového překryvu (obr. 4.23), přes desku a hlavní pouzdro, až k uchycení k držáku (obr. 4.24). Distance a zároveň upevnění jednotlivých vrstev zajišťují matice na šroubu natočené.



Obr. 4.21 : Držák zobrazovacího modulu



Obr. 4.22 : Centrální šroub procházející deskou budíků

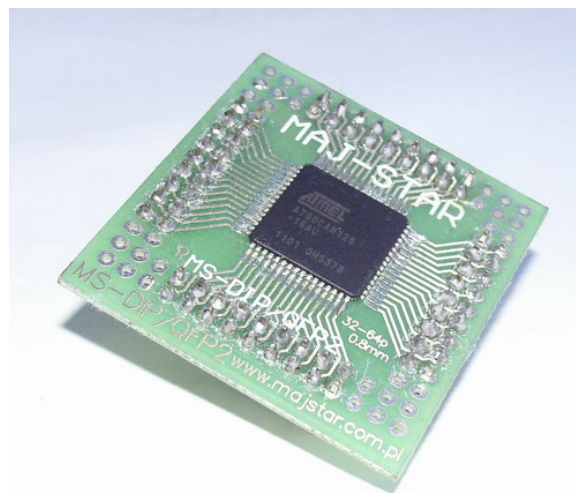


Obr. 4.23 : Překryv desky budíků



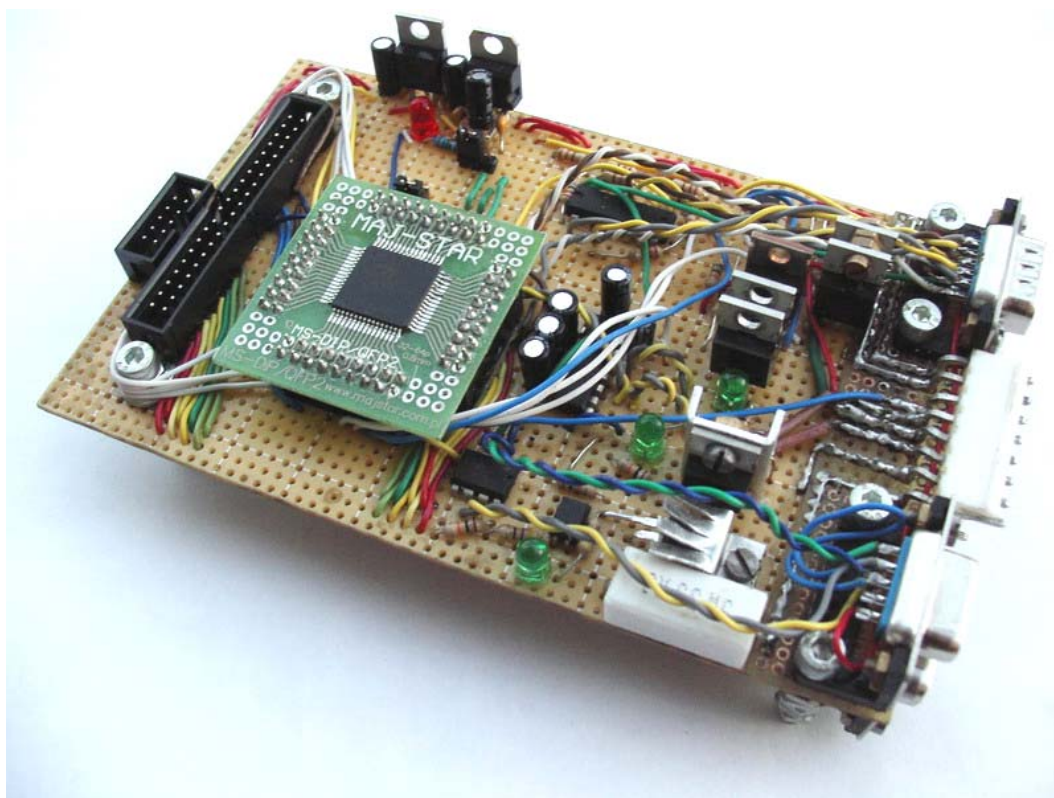
Obr. 4.24 : Zkompleťovaný modul budíků - uchycení

Následuje druhý díl skládačky, tedy centrální část jednotky. První problematika vyhotovení laboratorního prototypu vznikla díky volbě mikrokontroléru AT90128CAN a jeho neexistence v DIP pouzdru. Řešení je vyobrazeno na obr. 4.25, prostřednictvím zakoupené konverzní DPS, rozvádějící jednotlivé SMD vývody pouzdra TQFP64 na pady, odpovídající klasickému 2,54 mm rastru. Tím byl problém vyřešen a tvorba

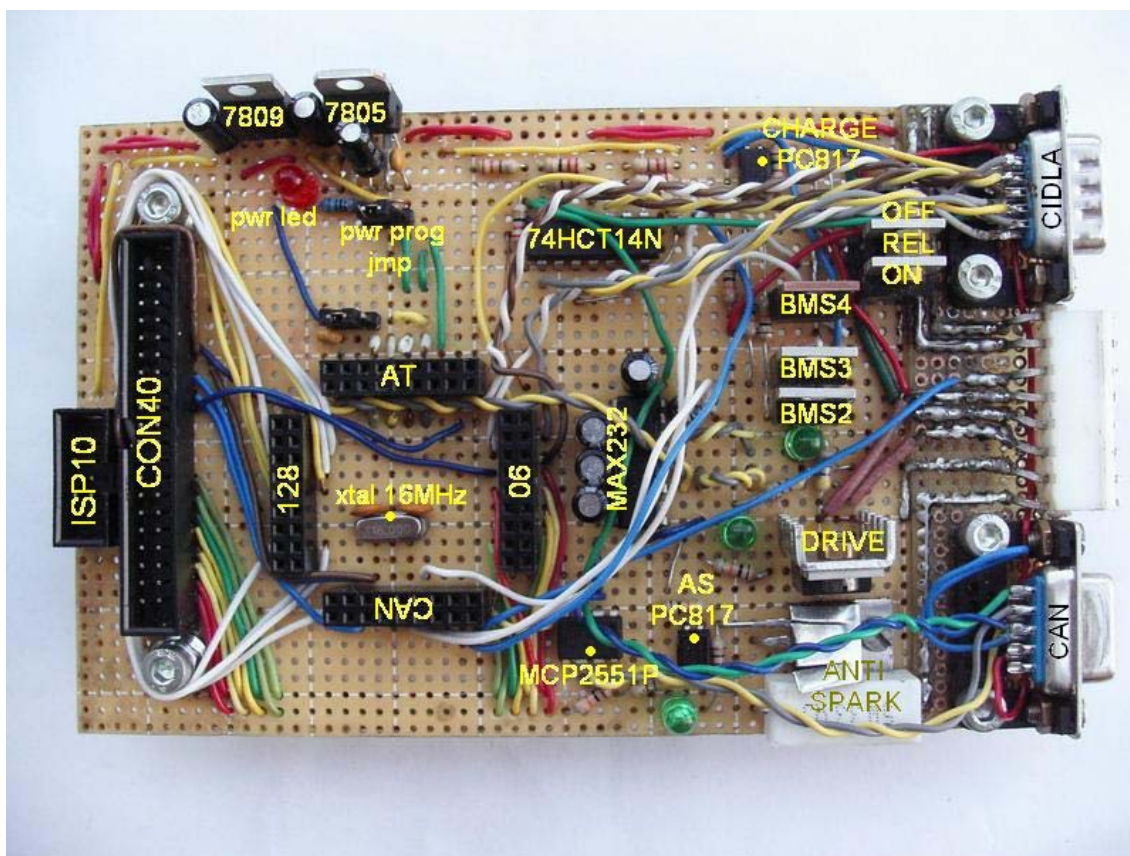


Obr. 4.25: AT90CAN128

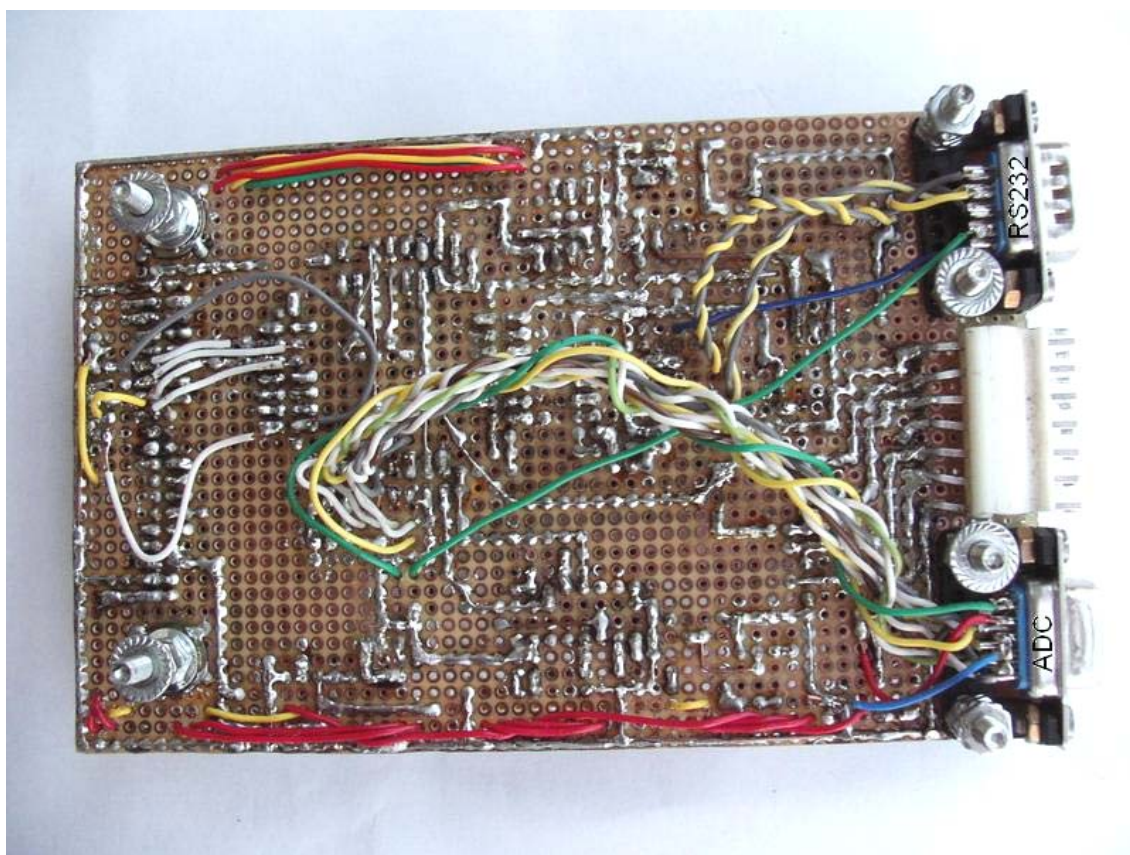
desky mohla započít. Několik desítek hodin pájení a zkoušení přineslo výsledek na obr. 4.26. Na následujícím obrázku 4.27 je fotografie desky s popisky osazení, obr. 4.28 vyobrazuje spodní stranu desky. Obr 4.29 je návodem, kam se zapojí jednotlivé ovládané a snímané periférie. Umístění jednotky na elektromobilu je zaznamenána na obr. 4.30, 4.31 a 4.32.



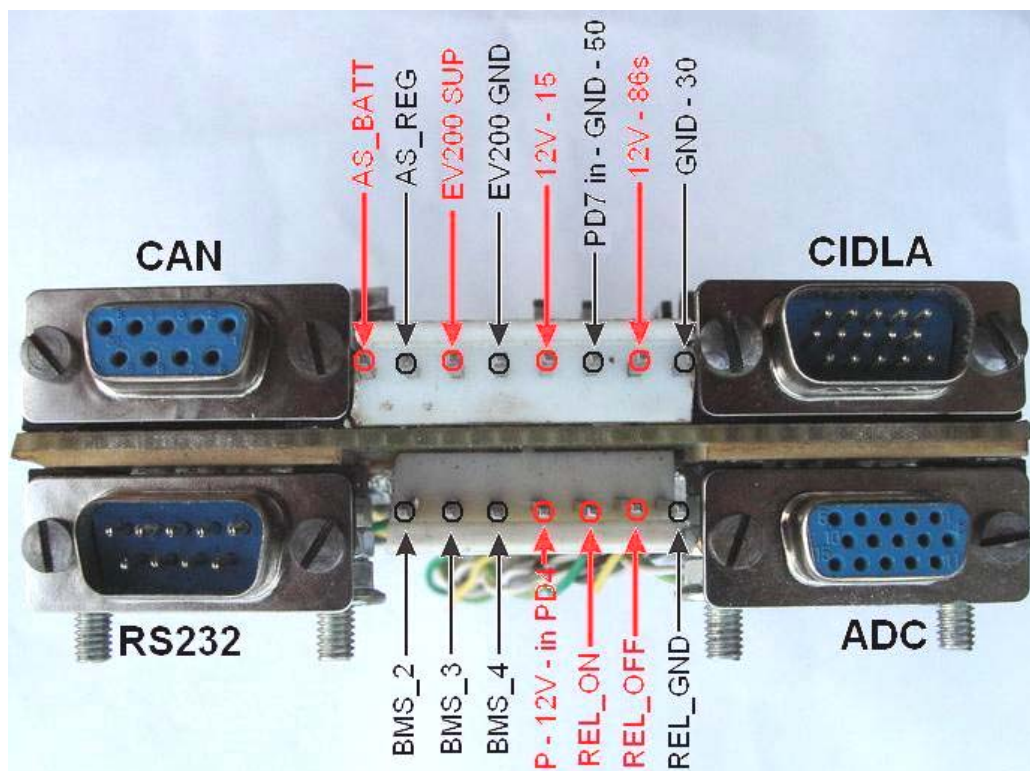
Obr. 4.26: Centrální část řídicí jednotky



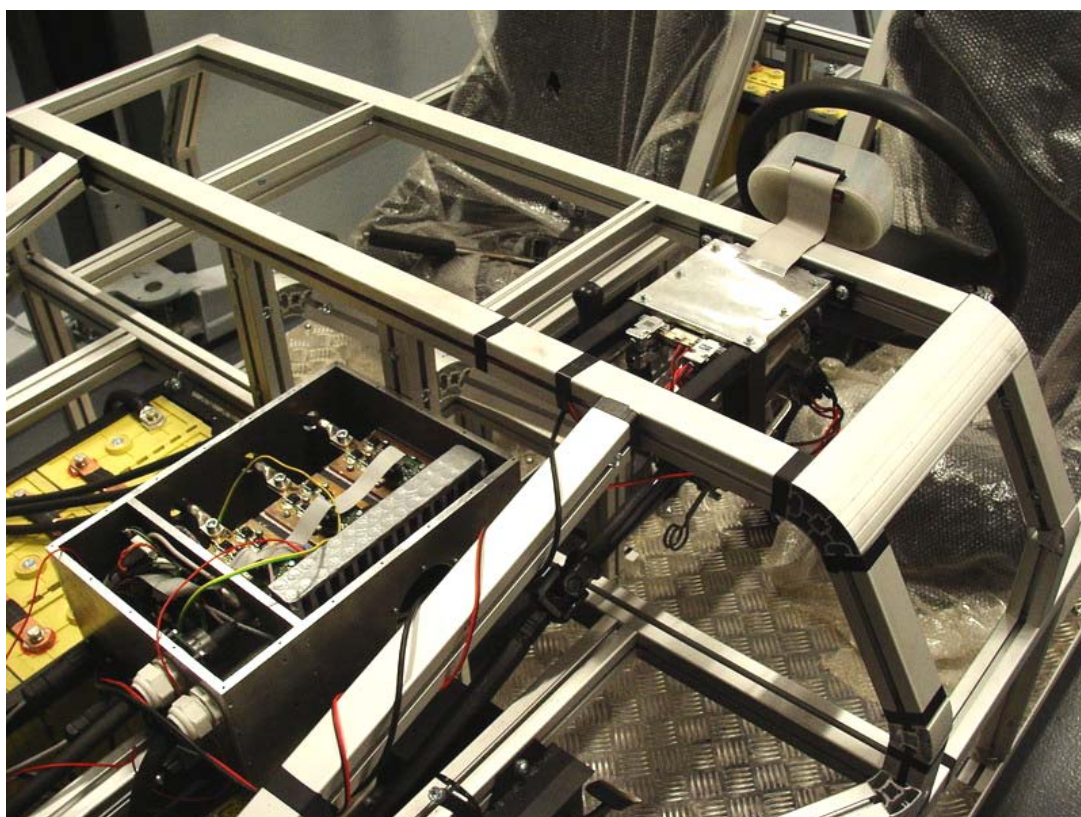
Obr. 4.27 : Laboratorní prototyp centrální části jednotky – osazení



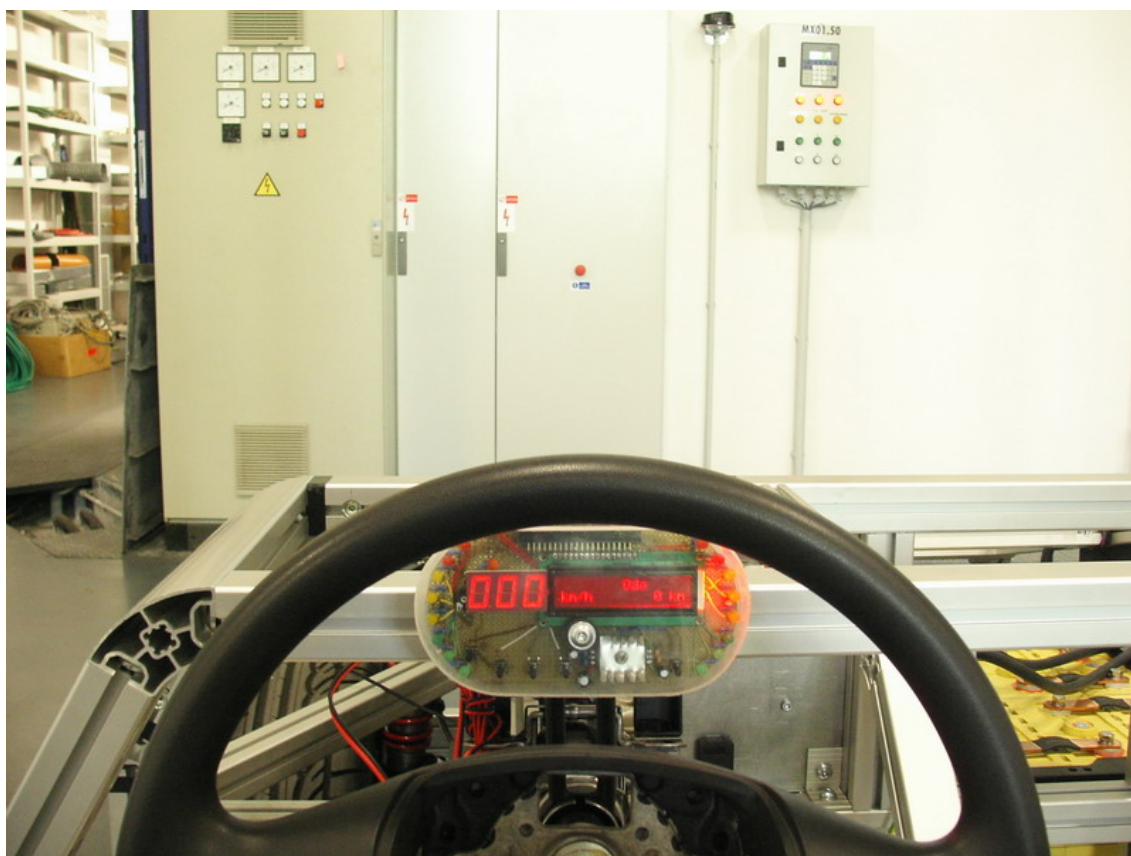
Obr. 4.28 : Laboratorní prototyp centrální části jednotky – spodní strana



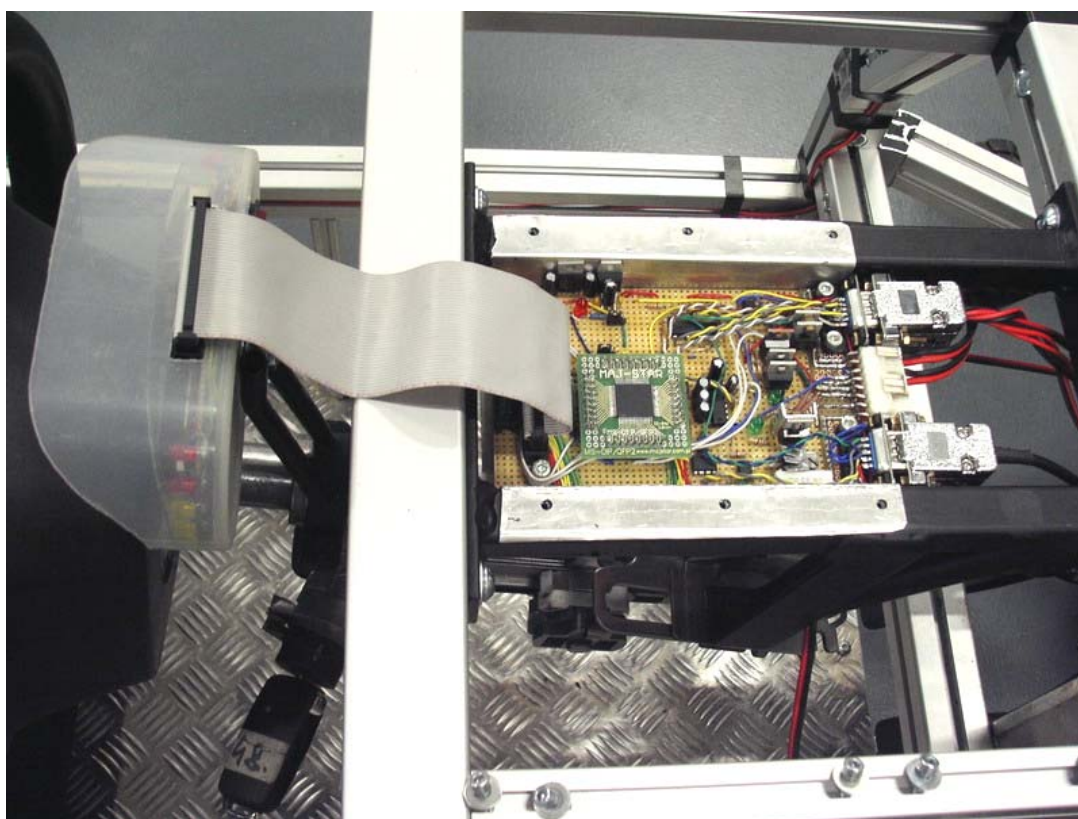
Obr. 4.29: Laboratorní prototyp centrální části jednotky – konektory



Obr. 4.30 : Řídicí jednotka instalovaná na elektromobilu



Obr. 4.31: Řidičův pohled na budíky

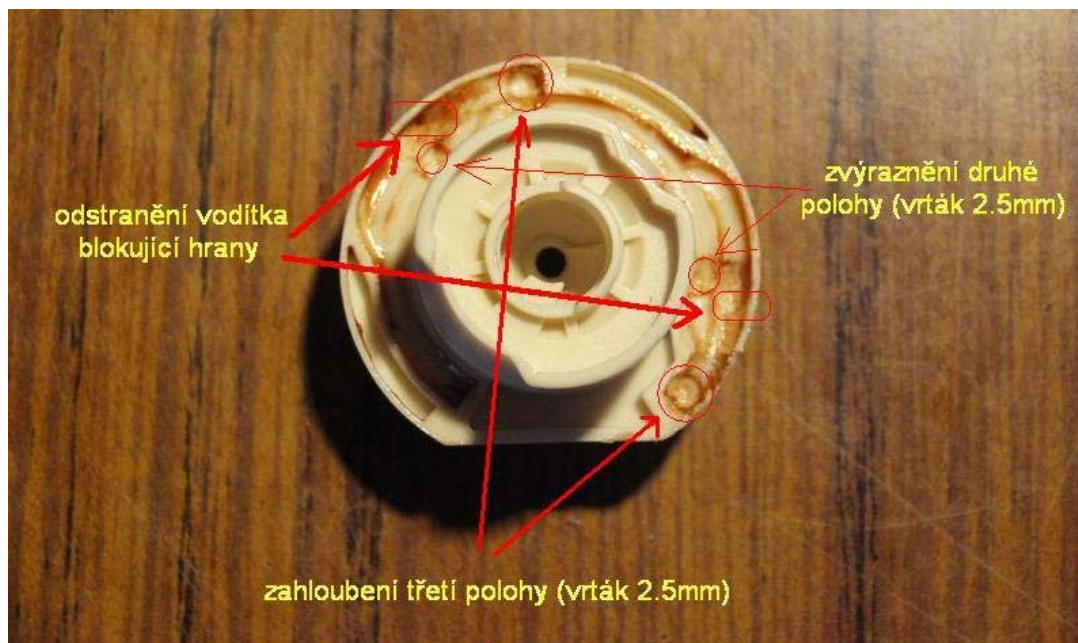


Obr. 4.32 : Řídicí jednotka bez vrchního krytu

4.9.2 Spínací skříňka

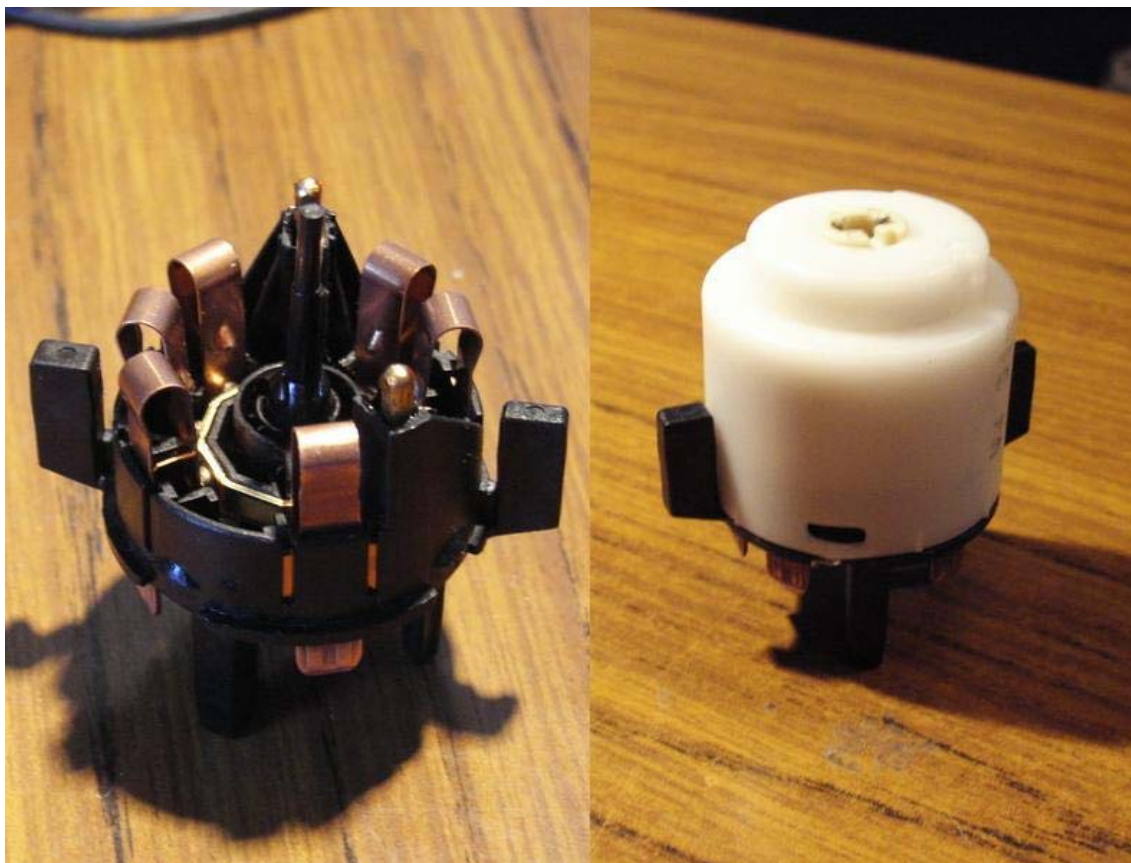
Jako poslední hardwarová záležitost zbývá už jen zpracování spínací skříňky. Zadáním bylo zapojit / upravit skříňku tak, aby provoz elektromobilu připomínal užívání běžného automobilu. Sekvence původní neupravené skříňky vypadala následovně. Zasunutí klíče před otočením žádnou událost nepřinese. Pootočení klíče z první polohy do druhé sepne napájení palubní elektroniky, v druhé poloze jsou aktivovány všechny řídicí okruhy běhu motoru a provozu auta. Otočením klíče do třetí polohy jsou aktivovány nezbytné okruhy a sepnut startér. Motor je nastartován a uvolníme-li klíč v třetí poloze, vrátí se nám vratnou pružinou zpět na druhou provozní polohu. Zde kulisa spínací skříňky blokuje možnost otočení klíče na třetí polohu. V případě potřeby opětovného startu je třeba vrátit se do polohy první a začít celou sekvenci znovu.

Nyní úprava pro užití v našem elektromobilu. Spínací skříňku je potřeba modifikovat tak, aby třetí poloha nebyla vratná a sepnutí v ní setrvalo, dále je třeba odstranit blokování přechodu z druhé do třetí polohy po návratu ze třetí polohy. Celá úprava kulisy je zachycena na obr. 4.33, vratná pružina určená k demontování je umístěna z druhé strany vyobrazeného tělesa.



Obr. 4.33 : Úprava kulisy spínací skříňky

Na obr. 4.34 je zachycen demontovaný modul spínání s pláštěm a bez pláště, obr. 4.35 zaznamenává pojmenování kontaktů korespondující se zhotovenými návrhy a schématy řídicí jednotky elektromobilu.



Obr. 4.34 : Demontovaný modul spínače skříňky



Obr. 4.35 : Kontakty spínací skříňky

- 12V napájecí zdroj se připojuje ke kontaktu 12, zem ke kontaktu 30

5. Gen 2.0 – software

Hardware je zkompletován, součástí této diplomové práce je i příprava zkušebního softwaru. Při výběru mikrokontroléru byl jeden z vítaných parametrů programovatelnost ve vyšším programovacím jazyce, v našem případě tedy jazyce C. Výhodou je, že se jedná o jeden z nejpoužívanějších programovacích jazyků, je k němu rozsáhlá podpora a mnoho známých vyřešených problematik.

5.1 Volba programovacího prostředí

Pro programovací jazyk C je k dispozici spousta vývojových prostředí. Naše řídicí jednotka je osazena MCU nesoucí označení AT90CAN128, založené na architektuře AVR od výrobce Atmel. Nejvhodnějším programovacím prostředím tak je AVRStudio, mnou používaná zastarávající, ale spolehlivá verze 4.19 bulid 730. Nahrávání programu do mikrokontroléru je prováděno přes program AVRdude miniaturním programátorem AVR USBasp, prostřednictvím ISP10.

5.2 Nastavení mikrokontroléru

Nastavení MCU začíná ještě před prvním naprogramováním. Řídicí jednotku jsme osadili externím krystalem a zároveň chceme využívat eeprom paměť. U rodiny AVR to znamená potřebu přepsání fuses (pojistek). Nová hodnota pojistek se musí dle požadavků vypočítat, snadno dostupný online kalkulačtor je například Engbedded Atmel AVR® Fuse Calculator [31].

5.3 Zdrojové kódy

Komentované zdrojové kódy hlavních částí programu a k nim náležící rutiny drivery a knihovny k nahlédnutí na přiloženém CD, jejich výčet a umístění na CD v dokumentu „Obsah.pdf“. Hlavní část zdrojového kódu softwaru v příloze D.

5.4 CAN – regulátor

Pro naše potřeby zatím postačující jednosměrná komunikace a to přijímání dat z regulátoru. Pakety jsou vysílány na linku periodicky, bez nutnosti vyžádání či potvrzení přijímací stranou. Rychlost sběrnice volitelná parametrem číslo 27, verze protokolu (adres) volitelná č. 35. Obsah jednotlivých packetů:

Obsah jednotlivých packetů:

Informační paket 1 – 7 datových bajtů

- adresa protokolu verze a – parametr č.15 (adresa)
- adresa protokolu verze b – $0x14A10000$ + parametr č.15
- napětí baterie – $(data[0] + 256*data[1])/57,45$ = napětí (V)
- proud z baterie – $(data[2] + 256*data[3])/10$ = proud (A)
- otáčky motoru – $(data[4] + 256*data[5] + 65536*data[6])*10$ = otáčky (ot/min)

Informační paket 2 – 7 datových bajtů

- adresa protokolu verze a – parametr č.15 (adresa) + 1
- adresa protokolu verze b – $0x14A10001$ + parametr č.15
- odometr (počet elektrických otáček od zapnutí)
– $(data[0] + 256*data[1] + 65536*data[2] + 16777216*data[3])$ = počet otáček
- teplota regulátoru – $(data[4])$ = teplota (st.C)
- teplota motoru – $(data[5])$ = teplota (st.C)
- teplota baterie – $(data[6])$ = teplota (st.C)

Informační paket 3 – 8 datových bajtů

- adresa protokolu verze a - parametr č.15 (adresa) + 2
- adresa protokolu verze b – $0x14A10002$ + parametr č.15
- vstupní (požadovaná) pwm – $(data[0] + 256*data[1])/10$ = požadovaná pwm (~%)
- výstupní (dodávaná) pwm – $(data[2] + 256*data[3])/10$ = dodávaná pwm (~%)
- varování (systém běží dál) – $(data[4] + 256*data[5])$ = rozklad bitů níže
 - $0x0001$ – podpětí
 - $0x0002$ – nadproud
 - $0x0004$ – přehřátí regulátoru
- chyba (systém odpojil) – $(data[6] + 256*data[7])$ = rozklad bitů níže
 - $0x0001$ – vstupní signál není/mimo rozsah
 - $0x0002$ – čeká se na průchod neutrálem
 - $0x0004$ – nutný reset - obecné
 - $0x0008$ – chyba interní paměti
 - $0x0010$ – nutný reset - změna nastavení

6. Závěr

Jaké jsou výsledky práce v porovnání se stanovenými cíly a zadáním. Tedy postupně. Tento projekt v sobě nesl od začátku mnoho otazníků a bylo velice těžké, nebo spíše nemožné, ho v době tvorby zadání jasně definovat a vytyčit. Známa k dispozici byl v době zadání pohon. Vše ostatní začalo dostávat konkrétní podobu až na přelomu zimního a letního semestru. Jak již bylo v úvodu práce zmíněno, projekt elektromobilu je společným projektem Fakulty strojní a Fakulty mechatroniky, ucelení požadavků a podkladů pro fyzickou realizaci řídicí jednotky vyžadovalo jasná stanoviska obou, co, kde a jak to bude potřeba zapojit. Zásadně ovlivňujícím prvkem, pro návrh a realizaci jednotky a palubní elektroniky, byla přednost konstrukce, bez větší možnosti ústupků. Hardware jednotky tak musel být namíru přizpůsoben danému prostoru.

Plnění cílů, nastolených zásadami pro vypracování diplomové práce, bylo rozptýleno mezi další, nové, v průběhu definování požadavků vzniklé, úkoly a požadavky nižších úrovní celého projektu. Pro upřesnění, zvolený a navrhnutý vhodný mikrokontrolér, schopný dle zásady pro vypracování č.3 komunikace s ostatními jednotkami po sběrnici CAN, naprogramovaný dle zásady č.4 ke sběru a zobrazení vybraných dat na LCD, zpracovávající je pro další jednotky a ukládající je pro pozdější zpracování je špatně použitelný, pakliže mu nepředchází aktivace ostatních jednotek a pohonu nutného pro vznik získávaných dat. Na bedra této diplomové práce tak spadlo několik nepříliš složitých úloh, nutných k vyřešení navíc, oproti zásadám pro vypracování. Mikrokontrolér tedy obsáhl krom zadaných cílů i ovládání aktivace a chodu dalších komponent tak, aby mu nemuseli být navrhovány další řídicí a aktivační prvky nižší úrovně už pro jeho pouhé testování.

Řídicí jednotka byla dle požadavků vyplývajících ze zadání a první analytické části práce navržena, sestrojena, instalována na elektromobil, zkoušena a její zkoušení a rozšiřování bude pokračovat i nadále, neb elektromobil je stále ve fázi vývoje.

Nyní několik dalších možností, jak postupovat dále ve vývoji jednotky, případně návrhu dalších generací jednotek. Zaprvé je to decentralizace. Jakýmsi módním trendem v oblastech vývoje automobilů je pro sebemenší drobnost vytvářet zvláštní řídicí jednotku. Užitečné to může být například pro oddělení senzorického systému od výpočetního, to samé platí pro systém zobrazovací a další, ale nic se nemá přehánět. Takový přístup ale vyžaduje návrh jasně definovaného komunikačního protokolu.

Další novinkou, kterou bych asi u dalšího návrhu zvažoval, by bylo využití platformy Arduino. Během této diplomové práce jsem při řešení jak hardwarových, tak softwarových problematik našel v online informačních zdrojích velké množství snadno řešených problematik. Již sama platforma, především co se týče vývoje softwaru, práci značně zjednodušuje a urychluje, a mnohým problematikám vůbec nedává vzniknout.

Dále mě napadá několik možných hardwarových vylepšení i současné řídicí jednotky, která byla vyvozena až při realizaci, zpětně do návrhu již nebyla zakomponována. Jako hlavní z nich bych vyzdvihl doplnění jednotky o obvod reálného času, pro zadávání parametru doby vytvoření souborů dat, zaznamenávaných na přenosnou paměť pro pozdější zpracování na PC.

Jakým závěrem práci ukončit. S pohledem do minulosti, kdy jsem se ze středoškolského všeobecného vzdělání díky technické vysoké škole dostal až k návrhu a následné úspěšné realizaci hardwaru i softwaru řídicí jednotky, a to ještě při vývoji elektromobilu, tj. tématu mě blízkému a tématu ve kterém vidím svoji budoucnost, bych řekl, že touto prací byly splněny nejen cíle jejího zadání.

Seznam použité literatury

- [1] Brian W. Kernighan, Dennis M. Ritchie: Programovací jazyk C, COMPUTER PRESS 2006 ISBN 978-80-2510-897-0
- [2] Matoušek David: Práce s mikrokontroléry Atmel AVR, BEN 2006 ISBN 80-7300-209-4
- [3] Matoušek David: Práce s inteligentními displeji LCD, BEN 2006 ISBN 80-7300-121-7
- [4] info GM EV1. [online]. [vid. 2013-05-14]. Dostupné z: <http://www.electroauto.cz/gm.html>
- [5] BLDC motor HCS C 40000W Industry. [online]. [vid. 2013-05-15]. Dostupné z: http://www.freeair.cz/freeaircz/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=173&category_id=52&option=com_virtuemart&Itemid=53
- [6] regulátor HBC 40063 [online]. [vid. 2013-05-15]. Dostupné z: http://www.freeair.cz/freeaircz/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=178&category_id=52&option=com_virtuemart&Itemid=53
- [7] Tyco Electronic: KILOVAC EV200. [online]. [cit. 2013-05-15]. Dostupné z: <http://relays.te.com/datasheets/ev200.pdf>
- [8] ilustrace. [online]. [vid. 2013-05-15]. Dostupné z: <http://www.ev-power.eu/>
- [9] Winston Battery: WB-LYP200AHA. [online]. [cit. 2013-05-15]. Dostupné z: http://en.winston-battery.com/index.php/products/power-battery/item/wb-lyp200ahab?category_id=176
- [10] GWL/POWER: POW48V30A Charger. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.i4wifi.cz/img.asp?attid=232585>
- [11] GWL/POWER: RL709-48V. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.i4wifi.cz/img.asp?attid=254771>
- [12] Aplikování sběrnice CAN. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.hw.cz/navrh-obvodu/rozhrani/aplikovani-sbornice-can.html>
- [13] ST Microelectronic: L7800 SERIES. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.gme.cz/dokumentace/330/330-004/dsh.330-004.1.pdf>

- [14] PALM TECHNOLOGY: plc0802aw. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.tme.eu/cz/Document/d5802ff3213b3a467cb2eff5f1daf157/plc0802aw-syl-1.pdf>
- [15] ATMEL: 8-bit Microcontroller ATmega8. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.gme.cz/dokumentace/432/432-027/dsh.432-027.1.pdf>
- [16] Diodes Incorporated: BZX55C3V3. [online]. [cit. 2013-05-15]. Dostupné z: <http://pdf1.alldatasheet.com/datasheet-pdf/view/58976/DIODES/BZX55C3V3.html>
- [17] PHILIPS: 74HCT14N. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.datasheetcatalog.org/datasheet2/7/0yu2erd9cyu9djs7ju4aqowj06cy.pdf>
- [18] Sériová linka RS232. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.hw.cz/rozhrani/hw-server-predstavuje-seriova-linka-rs-232.html#parametry>
- [19] ATMEL: 8-bit Microcontroller, ATmega644. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.gme.cz/dokumentace/432/432-222/dsh.432-222.1.pdf>
- [20] MICROCHIP: MCP2510I/P. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.gme.cz/dokumentace/433/433-329/dsh.433-329.1.pdf>
- [21] ATMEL: AT90CAN128. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.datasheetcatalog.org/datasheet2/7/0yu2erd9cyu9djs7ju4aqowj06cy.pdf>
- [22] ALLEGROMicroSystems: A1301. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.gme.cz/dokumentace/533/533-005/dsh.533-005.1.pdf>
- [23] SIEMENS: TLE4905G. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.gme.cz/dokumentace/533/533-447/dsh.533-447.1.pdf>
- [24] MICROCHIP: MCP2551-I/P. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.gme.cz/dokumentace/433/433-300/dsh.433-300.1.pdf>
- [25] MAXIM: MAX232CPE. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.gme.cz/dokumentace/433/433-100/dsh.433-100.1.pdf>
- [26] SHARP: optočlen PC817. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.gme.cz/dokumentace/523/523-130/dsh.523-130.1.pdf>
- [27] LITEON: LTM-8522HR. [online]. [cit. 2013-05-15]. Dostupné z: <http://elcodis.com/parts/764644/LTM-8522HR.html#datasheet>
- [28] Ye Mnie Mi Ho Mni: JHD162A. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.egochina.net.cn/eBay/Download/JHD162A.pdf>

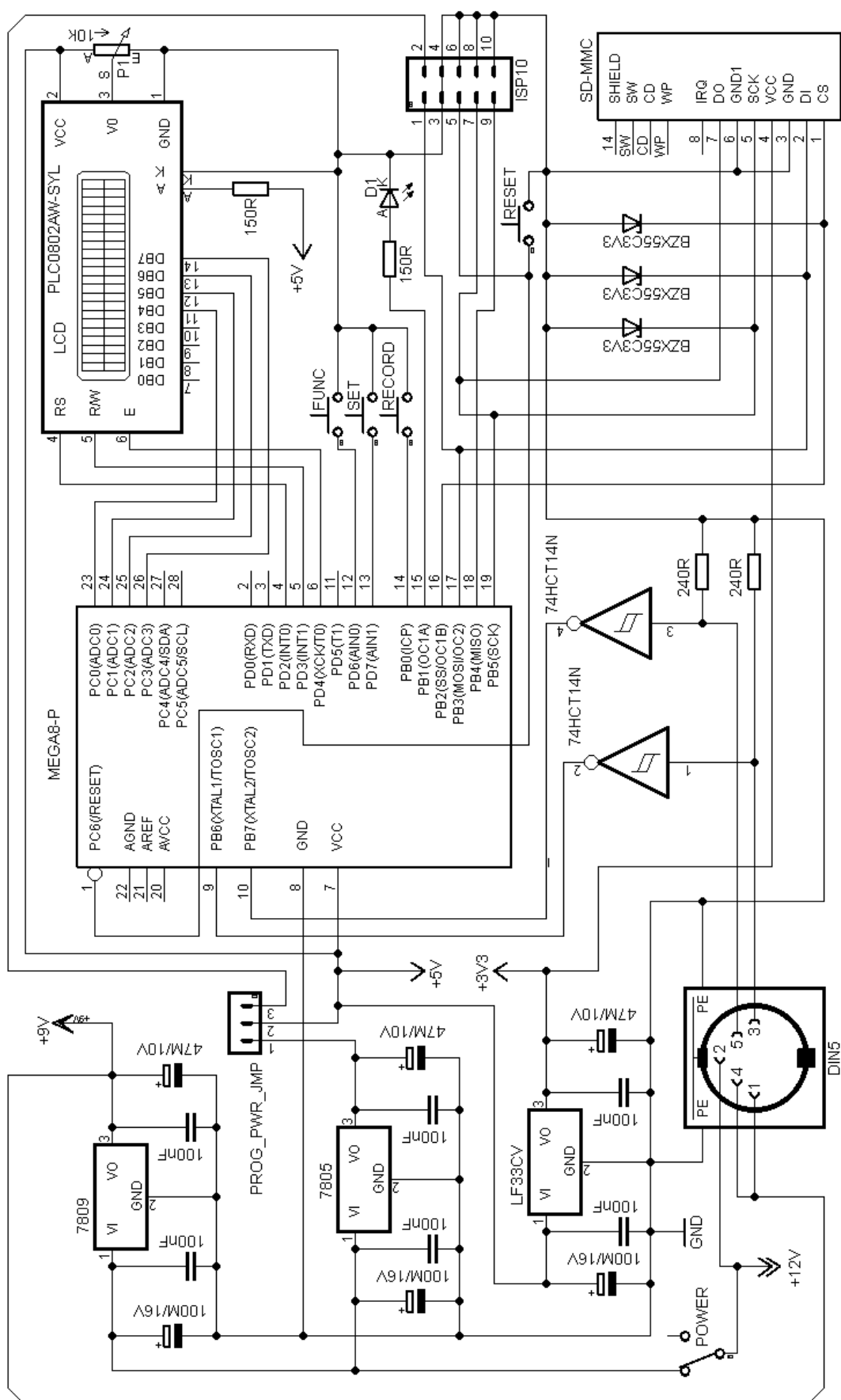
- [29] ST Microelectronics: LF33CV. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.gme.cz/dokumentace/330/330-165/dsh.330-165.1.pdf>
- [30] ilustrace SD/MMC. [online]. [cit. 2013-05-15]. Dostupné z: http://www.schripsema.org/pa3hdf/projects/cc1000/mmc_sd_contacts.jpg
- [31] Engbedded Atmel AVR® Fuse Calculator. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.engbedded.com/fusecalc>

Seznam Obrázků

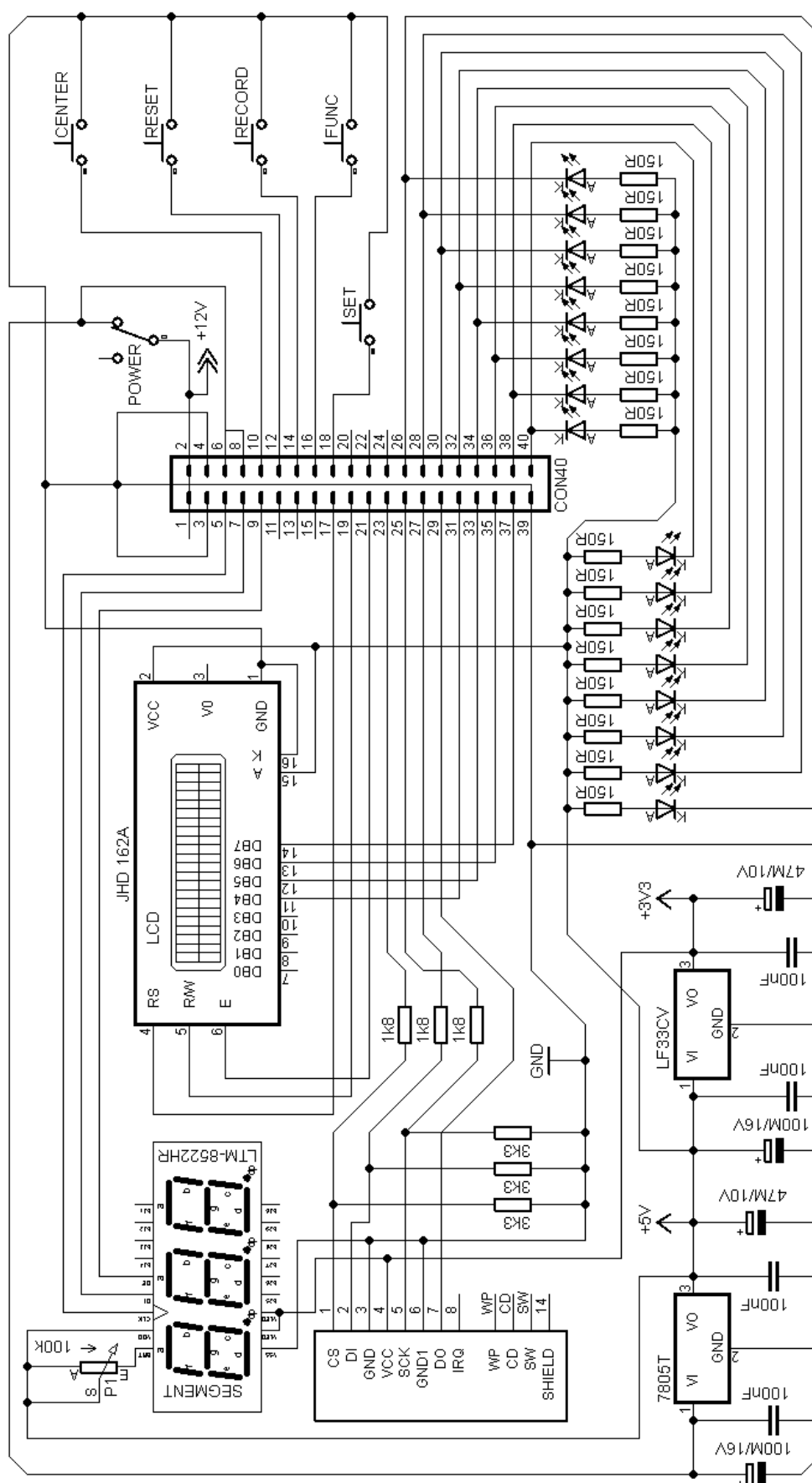
| | |
|--|----|
| Obr. 2.1 – Stavba elektromobilu e-Šus | 10 |
| Obr. 2.2 – BLDC motor HCS C 40000W Industry | 12 |
| Obr. 2.3 – Regulátor MGM Compro HBC-40063 | 12 |
| Obr. 2.4 – EV200 | 12 |
| Obr. 2.5 – 3.2V/200Ah | 12 |
| Obr. 2.6 – POW48V30A | 13 |
| Obr. 2.7 – relé RL709-48V | 13 |
| Obr. 2.8 – ABS sensor ATE | 13 |
| Obr. 2.9 – Magnetický kroužek | 13 |
| Obr. 2.10 – Spínací skříňka | 14 |
| Obr. 3.1 – Elektromobil s prototypem rychloměru 1.0 | 17 |
| Obr. 3.2 – Čidlo v pravé zadní těhlici | 18 |
| Obr. 3.3 – Snímač čidla vystupující z těhlice k magnetickému kroužku | 18 |
| Obr. 3.4 – Schéma zapojení rychloměru generace 1.0 | 20 |
| Obr. 4.1 – Blokové schéma | 23 |
| Obr. 4.2 – Vnitřní architektura AT90CAN128 | 27 |
| Obr. 4.3 – Rozmístění pinů na AT90CAN128 | 27 |
| Obr. 4.4 – Schéma zapojení MCU v řídicí jednotce | 28 |
| Obr. 4.5 – Nabíječ | 30 |
| Obr. 4.6 – Variace zapojení BMS konektoru nabíječe | 30 |
| Obr. 4.7 – Zapojení BMS konektoru nabíječe | 31 |
| Obr. 4.8 – Schéma zapojení obsluhy BMS konektoru nabíječe | 31 |
| Obr. 4.9 – Zapojení sériové linky s IO MAX232 | 32 |
| Obr. 4.10 – Schéma Antispark obvodu a spínacího tranzistoru relé EV200 | 33 |
| Obr. 4.11 – Principiální znázornění Anti-Spark obvodu | 34 |
| Obr. 4.12 – Schéma zapojení CAN | 35 |
| Obr. 4.13 – Senzorický systém | 36 |
| Obr. 4.14 – Snímač otáček elektromotoru | 37 |
| Obr. 4.15 – Schéma modulu budíků | 38 |
| Obr. 4.16 – LTM-8522HR | 38 |
| Obr. 4.17 – LCD JHD162A | 39 |
| Obr. 4.18 – Zapojení pinů SD/MMC rozhraní | 40 |

| | |
|--|----|
| Obr. 4.19 – Laboratorní prototyp zobrazovacího modulu – přední strana..... | 41 |
| Obr. 4.20 – Laboratorní prototyp zobrazovacího modulu – zadní strana..... | 42 |
| Obr. 4.21 – Držák zobrazovacího modulu..... | 43 |
| Obr. 4.22 – Centrální šroub procházející deskou budíků..... | 43 |
| Obr. 4.23 – Překryv desky budíků..... | 44 |
| Obr. 4.24 – Zkompletovaný modul budíků - uchycení..... | 44 |
| Obr. 4.25 – AT90CAN128..... | 45 |
| Obr. 4.26 – Centrální část řídicí jednotky..... | 45 |
| Obr. 4.27 – Laboratorní prototyp centrální části jednotky – osazení..... | 46 |
| Obr. 4.28 – Laboratorní prototyp centrální části jednotky – spodní strana..... | 46 |
| Obr. 4.29 – Laboratorní prototyp centrální části jednotky – konektory..... | 47 |
| Obr. 4.30 – Řídicí jednotka instalovaná na elektromobilu..... | 47 |
| Obr. 4.31 – Řidičův pohled na budíky..... | 48 |
| Obr. 4.32 – Řídicí jednotka bez vrchního krytu..... | 48 |
| Obr. 4.33 – Úprava kulisy spínací skříňky..... | 49 |
| Obr. 4.34 – Demontovaný modul spínače skříňky..... | 50 |
| Obr. 4.35 – Kontakty spínací skříňky..... | 50 |

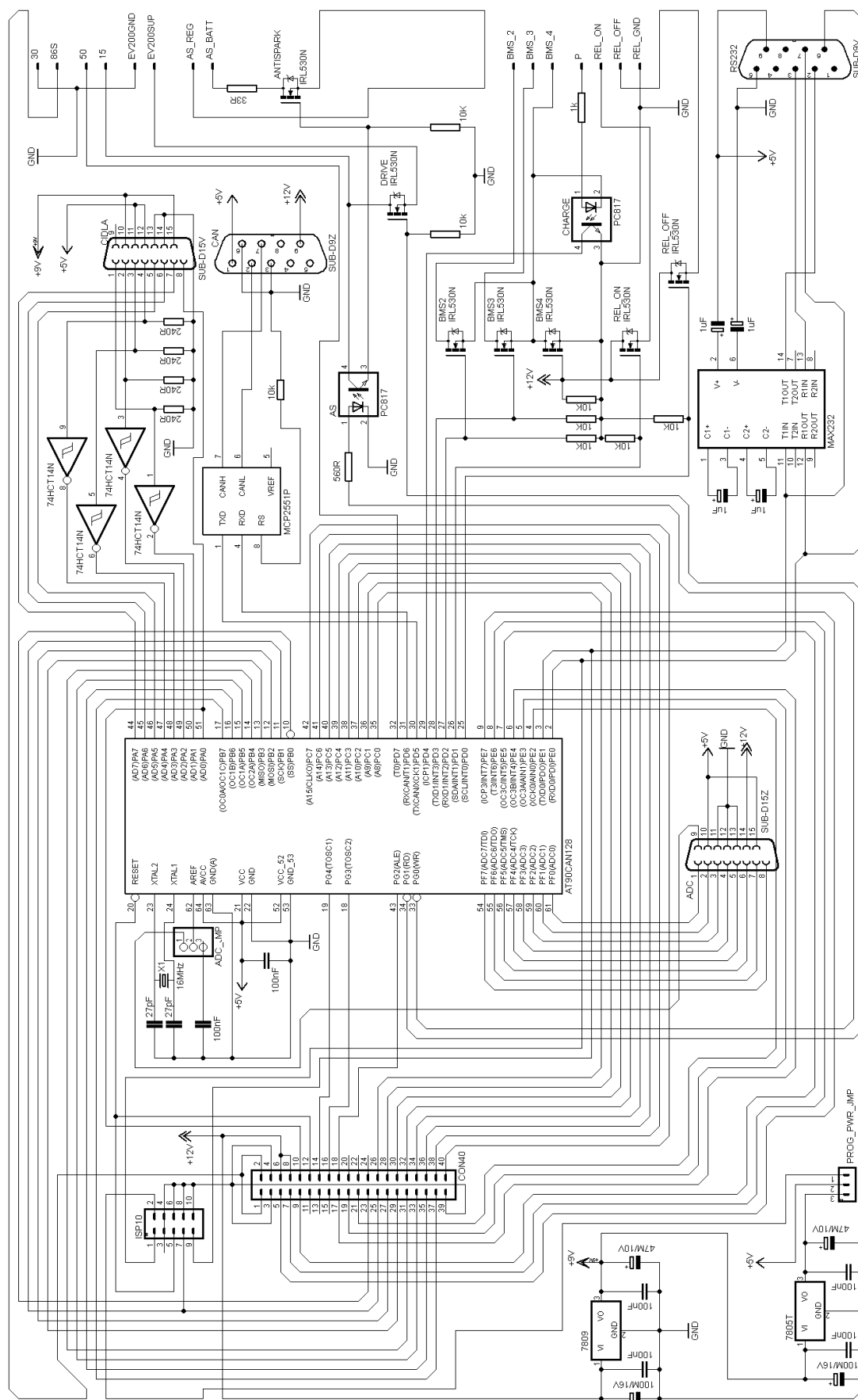
Příloha A - Schéma zapojení rychloměru 1.0



Příloha B - Schéma zapojení zobrazovacího modulu



Příloha C - Schéma zapojení centrálního modulu jednotky



Příloha D – Hlavní část zdrojového kódu

```
*****
This program was produced by the
AVR Studio 4.19 bulid 730
Project : Ridici_jednotka_elektromobilu
Version : 2.0
Date : 16.5.2013
Author : Puncman
Company : TUL
Chip type : AT90CAN128
Program type : Application
AVR Core Clock frequency: 16,000000 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 1024
*****/

#define F_CPU 16000000UL

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <avr/eeprom.h>
#include <avr/pgmspace.h>

#include <string.h>
#include <math.h>
#include "compat.h"
#include "lcd.h"
#include "uart.h"

#include "ff.h"
#include "diskio.h"

#include "can_drv.h"
#include "can_lib.h"

FATFS Fatfs[1];
FIL file1;

float EEMEM tot;
float EEMEM total;
uint8_t EEMEM mode;
string CANinfo;

int rychlost = 0;
int refrOt = 0;

float Distance;
float AllDistance;

int val[10][8] = {
    {1, 1, 1, 1, 1, 1, 0, 0},
    {0, 1, 1, 0, 0, 0, 0, 0},
    {1, 1, 0, 1, 1, 0, 1, 0},
    {1, 1, 1, 1, 0, 0, 1, 0},
    {0, 1, 1, 0, 0, 1, 1, 0},
    {1, 0, 1, 1, 0, 1, 1, 0},
    {1, 0, 1, 1, 1, 1, 1, 0},
    {1, 1, 1, 0, 0, 0, 0, 0},
}
```

```

        {1, 1, 1, 1, 1, 1, 1, 0},
        {1, 1, 1, 1, 0, 1, 1, 0}
};

int a;
int b;
int c;

//vypis cisla segmentovkou

void onSegment(int num) {
    a = num / 100;
    b = num / 10 - a * 10;
    c = num - a * 100 - b * 10;

    //cbi(PORTE,5);

    cbi(PORTE, 7);
    _delay_us(1);
    sbi(PORTE, 7);
    _delay_us(1);
    sbi(PORTE, 6);
    _delay_us(1);

    for (int i = 0; i < 8; i++) {
        cbi(PORTE, 7);
        _delay_us(1);
        sbi(PORTE, 7);
        _delay_us(1); //

        if (val[a][i] == 1) {
            sbi(PORTE, 6);
        } else {
            cbi(PORTE, 6);
        }
        _delay_us(1);
    }
    for (int i = 0; i < 8; i++) {
        cbi(PORTE, 7);
        _delay_us(1);
        sbi(PORTE, 7);
        _delay_us(1);
        if (val[b][i] == 1) {
            sbi(PORTE, 6);
        } else {
            cbi(PORTE, 6);
        }
        _delay_us(1);
    }

    for (int i = 0; i < 8; i++) {
        cbi(PORTE, 7);
        _delay_us(1);
        sbi(PORTE, 7);
        _delay_us(1); //

        if (val[c][i] == 1) {
            sbi(PORTE, 6);
        } else {
            cbi(PORTE, 6);
        }
    }
}

```



```

        _delay_us(1);
    }

    for (int i = 0; i < 11; i++) {
        cbi(PORTE, 7);
        _delay_us(1);
        sbi(PORTE, 7);
        _delay_us(1);
    }
}

void distanceIncr() {
    Distance = Distance + 0.1;
    eeprom_write_float(&tot, Distance);
}

void allDistanceIncr() {
    AllDistance = AllDistance + 0.1;
    eeprom_write_float(&total, AllDistance);
}

void ClearScreen() {
    SendString("                ", 1, 0);
    SendString("                ", 2, 0);
}
int sekvence = 0;
int sekvenceZap = 0;

void runCard() {
    disk_initialize((BYTE) 0);
    SendString("init", 1, 0);
    f_mount((BYTE) 0, &Fatfs[0]);
    SendString("mount", 1, 0);
    f_open(&file1, "File1.txt", FA_WRITE | FA_CREATE_ALWAYS);
    f_puts("data1 = 0x11;\n", &file1);
    f_puts("data2 = 0x22;\n", &file1);
    f_puts("data3 = 0x44;\n", &file1);
    f_close(&file1);
}

int main(void) {

    Distance = eeprom_read_float(&tot); //0
    AllDistance = eeprom_read_float(&total); //0

    int traillong = 0;
    int blik = 0;
    int refr = 0;
    int mod = eeprom_read_byte(&mode);

    int touch1 = 0;
    int touch2 = 0;
    int touch3 = 0;
    int touched = 0;

    int spin1 = 0;
    int spin2 = 0;

```

```

int chain1 = 0;
int chain2 = 0;

DDRA = 0x00; //pull up na port B
DDRG = 0x00; //pull up na port D

sbi(DDRG, 0);
sbi(DDRG, 1);
cbi(DDRD, 7);

DDRC = 0xff;

PORTA = 0xff;
PORTG = 0xff;
PORTD = 0xff;

sbi(DDRE, 5);
cbi(PORTE, 5);
sbi(DDRE, 6);
sbi(DDRE, 7);

InitLCD();
Init_MMC();
Init_Can();
_delay_ms(77);

SendString(" e-Sus          ", 1, 0);
SendString("          Electric ", 2, 0);

int j = 0;
for (int i = 0; i < 230; i++) {

    if (i < 115)
        j++;
    else {
        j--;
    }
    if (j > 12)
        cbi(PORTC, 0);
    else sbi(PORTC, 0);
    if (j > 25)
        cbi(PORTC, 1);
    else sbi(PORTC, 1);
    if (j > 50)
        cbi(PORTC, 2);
    else sbi(PORTC, 2);
    if (j > 62)
        cbi(PORTC, 3);
    else sbi(PORTC, 3);
    if (j > 75)
        cbi(PORTC, 4);
    else sbi(PORTC, 4);
    if (j > 87)
        cbi(PORTC, 5);
    else sbi(PORTC, 5);
    if (j > 100)
        cbi(PORTC, 6);
    else sbi(PORTC, 6);
    if (j > 112)
        cbi(PORTC, 7);
    else sbi(PORTC, 7);
}

```

```

        onSegment(j);
        _delay_us(100);
    }

    int otacky = 0;
    int spinVal = 0;
    int value = 255;

    while (1) {

        //-----
        //obnova displaye, segmentovky a otacek a provadeni vypoctu
        kazdych 500ms;

        if (refr > 500) {
            if (spinVal >= 32)
                PORTC = 0;
            else if (spinVal >= 28)
                PORTC = 128;
            else if (spinVal >= 24)
                PORTC = 192;
            else if (spinVal >= 20)
                PORTC = 224;
            else if (spinVal >= 16)
                PORTC = 240;
            else if (spinVal >= 12)
                PORTC = 248;
            else if (spinVal >= 8)
                PORTC = 252;
            else if (spinVal >= 4)
                PORTC = 254;
            else PORTC = 255;
            spinVal = 0;

            mod = eeprom_read_byte(&mode);
            traillong = traillong + chain1 + chain2;

            if (traillong > 4494) {
                traillong = traillong % 4494;
                distanceIncr();
                allDistanceIncr();
            }
            spinVal = 0;
            rychlost = (chain1 + chain2) * 16 / 100;
            refr = 0;
            if (touched > 0 && touched < 10) {
                touched = touched - 1;
            } else {
                ClearScreen();
                //SendInt(rychlost, 1, 0);
                if (mod == 1) {
                    SendString("      Odo      ", 1, 0);
                    SendFloatZero(AllDistance, 2, 8);
                } else {
                    SendString("      Dist      ", 1, 0);
                    SendFloat(Distance, 2, 8);
                }
                SendString("km/h", 2, 0);
                //SendString("km/h", 1, 4);
                SendString("km", 2, 14);
            }
        }
    }

```

```

    }
    chain1 = 0;
    chain2 = 0;

    onSegment(rychlost);
    CANinfo = onCANGenInfo();
    onMMCWrite(CANinfo);

} else {
    refr++;
}

//-----
//func tlacitko
//po min 100ms stisknuti vykonani akce
//po min 3000ms stisknuti vykonani druhe akce

if (PING & 0b00001000) { //neni stisknuto
    if (touch1 > 3000) {
        touch1 = 0;
        touched = 2;
    } else if (touch1 > 100) {
        touch1 = 0;
        if (eeprom_read_byte(&mode) == 0) {
            eeprom_write_byte(&mode, 1);
            SendString("      Odo      ", 1, 0);
            SendFloatZero(AllDistance, 2, 8);
        } else {
            eeprom_write_byte(&mode, 0);
            SendString("      Dist      ", 1, 0);
            SendFloat(Distance, 2, 8);
        }
        touched = 2;
    }
} else {
    touch1++;

    if (touch1 > 3000 && touch1 < 3002) {
        SendString("      cleared      ", 1, 0);
        if (mod == 1) {
            SendFloatZero(0.0f, 2, 0);
            AllDistance = 0.0f;
            eeprom_write_float(&total, AllDistance);
        } else if (mod == 0) {
            SendFloat(0.0f, 2, 0);
            Distance = 0.0f;
            eeprom_write_float(&tot, Distance);
        }
        touch1 = 3002;
        touched = 2;
    } else if (touch1 > 2000 && touch1 < 2002) {
        SendString("      clearing      ", 1, 0);
        if (mod == 1) SendFloat(AllDistance, 2, 8);
        else SendFloat(Distance, 2, 8);
        touched = 2;
    }
}

//-----
//tlacitko set (testovaci usecase)
//po min 100ms stisknuti vykonani akce

```

```

if (PING & 0b00000100) { //neni stisknuto
    if (touch2 > 100) {
        touch2 = 0;
        allDistanceIncr();
        SendString(" Increment Odo ", 1, 0);
        SendFloatZero(AllDistance, 2, 8);
        touched = 2;
    }
} else {
    touch2++;
    if (touch2 > 100) {
        touch2 = 101;
    }
}

//-----
//tlacitko center (testovaci usecase)
//po min 100ms stisknuti vykonani akce

if (PINA & 0b00000001) { //neni stisknuto
    if (touch3 > 100) {
        touch3 = 0;
        distanceIncr();
        SendString(" Increment Dist ", 1, 0);
        SendFloat(Distance, 2, 8);
        touched = 2;
    }
} else {
    touch3++;
    if (touch3 > 100) {
        touch3 = 101;
    }
}

//-----
//cidlo 1
//chain1 inkrementovane s 0->1

if (PINA & 0b00000010) {
    if (spin1 == 0) {
        chain1++;
    }
    spin1 = 1;
} else {
    spin1 = 0;
}

//-----
//cidlo 2
//chain2 inkrementovane s 0->1

if (PINA & 0b00000100) {

    if (spin2 == 0) {
        chain2++;
    }
    spin2 = 1;
} else {
    spin2 = 0;
}

```

```

    }

    //-----
    // cidlo otacek elektromotoru
    // hallova sonda

    if (PIN_A & 0b00100000) {
        if (otacky == 0) {
            spinVal++;
        }
        otacky = 1;
    } else {
        otacky = 0;
    }

    //-----
    //sekvence Antispark/Start
    //3ti poloha klince, 3s antispark, hlavni rele

    if (PIND & 0b10000000) {
        sekvenceZap = 0;
        sekvence = 0;
        cbi(PORTG, 0);
        cbi(PORTG, 1);
    } else {
        if (sekvenceZap == 0) {
            sekvenceZap = 1;
            sbi(PORTG, 0);
            SendString("AntiSpark", 1, 0);
            SendString("charging", 2, 0);
            touched = 7;
        }
        if (sekvenceZap == 1) {
            if (sekvence <= 3500)
                sekvence = sekvence + 1;
        }
        if (sekvence >= 3000) {
            sbi(PORTG, 1);
            if (sekvence == 3300) {
                sekvence = 3301;
                cbi(PORTG, 0);
                SendString("Ready", 1, 0);
                SendString("to GO !!!", 2, 0);
                touched = 2;
            }
        }
    }
    _delay_ms(0.125);
}

```